Approximate Problems for Finite Transducers

Emmanuel Filiot
Université libre de Bruxelles, Belgium

Ismaël Jecker
Université Marie et Louis Pasteur, CNRS, institut FEMTO-ST, F-25000 Besançon, France

Khushraj Madnani
Max Planck Institute for Software Systems (MPI-SWS), Saarbrücken, Germany

Saina Sunny
Indian Institute of Technology Goa, India

Abstract

Finite (word) state transducers extend finite state automata by defining a binary relation over finite words, called *rational relation*. If the rational relation is the graph of a function, this function is said to be rational. The class of sequential functions is a strict subclass of rational functions, defined as the functions recognised by input-deterministic finite state transducers. The class membership problems between those classes are known to be decidable. We consider approximate versions of these problems and show they are decidable as well. This includes the *approximate functionality problem*, which asks whether given a rational relation (by a transducer), is it *close* to a rational function, and the *approximate determinisation problem*, which asks whether a given rational function is close to a sequential function. We prove decidability results for several classical distances, including Hamming and Levenshtein edit distance. Finally, we investigate the *approximate uniformisation problem*, which asks, given a rational relation R, whether there exists a sequential function that is close to some function uniformising R. As its exact version, we prove that this problem is undecidable.

2012 ACM Subject Classification Theory of computation → Transducers; Theory of computation → Quantitative automata

Keywords and phrases Finite state transducers, Edit distance, Determinisation, Functionality

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.155

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

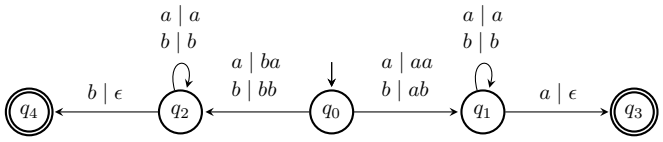
Related Version *Full Version*: `https://arxiv.org/abs/2504.17299`

Funding *Emmanuel Filiot*: He is a senior research associate at the National Fund for Scientific Research (F.R.S.-FNRS) in Belgium. His work is supported by the FNRS project T011724F.

1 Introduction

Finite (state) transducers are a fundamental automata model to compute functions from words to words. The literature on finite state transducers is rich, and dates back to the early days of computer science, where they were called generalised sequential machines [29, 20]. See also [28, 19] and the references therein. Finite state transducers extend finite automata with outputs on their transitions, allowing them to produce none or several symbols. While finite automata recognise languages of (finite) words, finite transducers recognise binary relations from words to words, called *rational relations*. When the rational relation is the graph of a function, it is called a *rational function*. This subclass is decidable within the class of rational relations. In particular, given a finite transducer T, it is decidable in PTIME whether T recognises a function [21, 10]. In that case, T is said to be *functional*. Beyond the fact that it is a natural restriction, the class of functional transducers is of high importance, as many problems known to be undecidable for transducers (such as inclusion and equivalence), become decidable under the functional restriction.


Determinisation. It turns out that non-determinism is needed for finite transducers to capture rational functions. A canonical example is the function $f_{\mathsf{last}} : \{a, b\}^* \to \{a, b\}^*$ which moves the last symbol upfront. For example, $f_{\mathsf{last}}(ab\underline{b}) = \underline{b}ab$ and $f_{\mathsf{last}}(ab\underline{a}) = \underline{a}ab$. Since the number of symbols that have to be read before reading the last symbol is arbitrarily large, a finite transducer recognising f_{last} needs non-determinism to guess the last symbol, as illustrated by the following transducer:



So non-determinism, unlike finite automata, brings some extra expressive power when it comes to finite transducers. On the other hand, non-determinism also yields some inefficiency issues when the input is received sequentially as a stream, because the whole input may have to be stored in memory until the first output symbol can be produced. This motivates the class of *sequential functions*, as the rational functions recognised by *input-deterministic* finite transducers, and the *determinisation problem*: given an arbitrary finite transducer, does it recognise a sequential function? In other words, can it be (input-) determinised? This well-studied problem is known to be decidable in PTime [10]. The determinisation problem is a central problem in automata theory. It has been for instance extensively considered for weighted automata [27], and a long-standing open problem is whether this problem is decidable for $(\mathbb{N}, max, +)$-automata [25].

Approximate determinisation. The function f_{last} is not sequential, in other words, the latter transducer is not determinisable. It turns out that f_{last} is *almost* sequential, in the sense that it is *close to* some sequential function, for instance the identity function id. "Close to" can be defined in different ways, by lifting standard distances between words to functions and relations. Two classical examples are the Hamming distance and the Levenshtein distance, which respectively measure the minimal number of letter substitutions (respectively letter substitutions, insertion and deletion) to rewrite a word into another. A distance d between words is lifted to functions f_1, f_2 with the same input domain, by taking the supremum of $d(f_1(u), f_2(u))$ for all words u in their domain. Coming back to our example, f_{last} and id are close for the edit distance, in the sense that $d(f_{\mathsf{last}}, \mathsf{id})$ is finite for d the edit distance, but they are not close for the Hamming distance. This raises a natural and fundamental problem, called *approximate determinisation* problem (for a distance d): given a finite transducer recognising a function f, does there exists a sequential function s such that $d(f, s)$ is finite? The approximate determinisation problem has been extensively studied for weighted automata [4, 8, 5] and quantitative automata [6, 7], but, to the best of our knowledge, nothing was known for transducers. However, if both f and s are given (by finite transducers), checking whether they are close (for various and classical edit distances) is known to be decidable, even if s is rational but not sequential [2]. This can be seen as the verification variant of approximate determinisation, while approximate determinisation is rather a synthesis problem, for which only f is given, and which asks to generate s if it exists.

Contributions. In this paper, our main result is the decidability of approximate determinisation of finite transducers, for a family $\mathcal{E}$ of edit distances, which include Hamming and Levenshtein distances. For exact determinisation, determinisable finite transducers are characterised by the so called twinning property (**TP**) [10, 9, 13], a pattern that requires

that the delay between any two outputs on the same input must not increase when taking synchronised cycles of the transducer. As noticed in [2], bounded (Levenshtein) edit distance is closely related to the notion of word conjugacy. In this paper, we consider an approximate version of the twinning property (**ATP**), with no constraints on the delay, but instead requires that the output words produced on the synchronised loops are conjugate. It turns out that **ATP** is not sufficient to characterise approximately determinisable transducers, and an extra property is needed, the strongly connected twinning property (**STP**), which requires that the **TP** holds within SCCs of the finite transducer. We show that a transducer $\mathcal{T}$ is approximately determinisable (for Levenshtein distance) iff both **ATP** and **STP** hold and, if they do, we show how to approximately determinise $\mathcal{T}$. We also prove that **ATP** and **STP** are both decidable.

For Hamming distance, which only allows letter substitutions, determinizable transducers are characterized by both **STP** and another property called Hamming twinning property (**HTP**), which roughly requires that outputs on synchronised cycles have same length and do not mismatch. We also show that **HTP** is decidable, which entails decidability of approximate determinisation for Hamming distance.

We also consider another fundamental problem: the *approximate functionality problem*. Informally, the approximate functionality problem asks whether a given rational relation R is *almost* a rational function, in the sense that $d(R, f)$ is finite for some rational function f, where $d(R, f)$ is now the supremum, for all $(u, v) \in R$, of $d(v, f(u))$. We prove that the approximate functionality problem is decidable for all the distances in $\mathcal{E}$. We prove this problem to be decidable for classical distances, including Hamming and Levenshtein.

Finally, we consider the *approximate (sequential) uniformisation problem*. In its exact version, this problem asks whether for a given rational relation R, there exists a sequential function f with the same domain, and whose graph is included in R. This problem is closely related to a synthesis problem, but is unfortunately undecidable [11, 17]. We consider its approximate variant, where instead of requiring that the graph of f is included in R, we require that it is close to some function whose graph is included in R. However, despite this relaxation, we show that the problem is still undecidable. The proofs omitted are provided in the full version.

Other related works. Variants of the determinisation problem have been considered in the literature [17]. However, this work considers the *exact* determinisation of a transducer $\mathcal{T}$, by some sequential transducer which is *close* to $\mathcal{T}$, for some notions of structural similarity between transducers. Robustness and continuity notions for finite transducers have been introduced in [22]. While those notions are also based on word distances, the problems considered are different from ours.

2 Preliminaries

For every $k \in \mathbb{N}$, let $[k]$ denote the set $\{1, 2, \ldots, k\}$, and for integers $i \leq j$, let $[i, j]$ denote the set $\{i, i+1, \ldots, j\}$.

Words. Let A or B denote finite alphabets of letters. A word is a sequence of letters. The empty word is denoted by ϵ. The length of a word is denoted by $|w|$, in particular $|\epsilon| = 0$. The ith letter of a word w, for $i \in \{1, \ldots, |w|\}$, is denoted by $w[i]$. The primitive root of w is the shortest word ρ_w such that $w \in \rho_w^*$. The set of all finite words over the alphabet A is denoted by A^*. A relation $R \subseteq A^* \times B^*$ is sometimes called a *transduction*, and is said to be

functional if it is the graph of a function. We let $dom(R) = \{u \in A^* \mid \exists v \in B^*, (u,v) \in R\}$ be the *domain* of R, and for all $u \in A^*$, we let $R(u) = \{v \in B^* \mid (u,v) \in R\}$. Note that $u \in dom(R)$ iff $R(u) \neq \varnothing$.

Finite State Automata and Transducers. A (non-deterministic) finite automaton over an alphabet A is denoted as a tuple $\mathcal{A} = (Q, I, \Delta, F)$ where Q is the finite set of states, $I \subseteq Q$ the set of initial states, $F \subseteq Q$ the set of final states, and $\Delta \subseteq Q \times A \times Q$ the transition relation. A *run* on a word $\sigma_1 \dots \sigma_n$ is a sequence of states $q_1 \dots q_{n+1}$ such that (q_i, σ_i, q_{i+1}) for all $i \in [n]$. It is accepting if and only if $q_1 \in I$ and $q_{n+1} \in F$. We denote by $L(\mathcal{A})$ the language accepted by $\mathcal{A}$, i.e. the set of words on which there is an accepting run. An automaton is said to be *trim* if for any state q, there exists at least one accepting run visiting q. When $\mathcal{A}$ is deterministic, we denote the transition function as $\delta : Q \times A \to Q$.

A *rational transducer* $\mathcal{T}$ over an input alphabet A and an output alphabet B is a (non-deterministic) finite automaton over a finite subset of $A^* \times B^*$. A transition $(p, (u,v), q)$ of $\mathcal{T}$ is often denoted $p \xrightarrow{u|v}_{\mathcal{T}} q$. More generally, we write $p \xrightarrow{u_1 \dots u_n | v_1 \dots v_n}_{\mathcal{T}} q$ whenever there exists a run ρ of $\mathcal{T}$ from p to q on $(u_1, v_1) \dots (u_n, v_n)$. The *relation recognised* by $\mathcal{T}$, denoted as $R_{\mathcal{T}} \subseteq A^* \times B^*$, is defined as $R_{\mathcal{T}} = \{(u,v) \mid \exists q_0 \in I, q_f \in F, q_0 \xrightarrow{u|v}_{\mathcal{T}} q_f\}$. Relations recognised by rational transducers are called *rational relations*.

When rational transducers recognise functions, it is often convenient to restrict their transitions to input words of length one exactly, modulo the possibility of producing a word on accepting states. This defines the so-called class of real-time transducers, which is expressively equivalent to rational transducers when restricted to functions. Formally, a *real-time transducer* (or simply, a transducer in the sequel) $\mathcal{T}$ over input alphabet A and output alphabet B is given by a tuple $\mathcal{T} = (Q, I, \Delta, F, \lambda)$ where (Q, I, Δ, F) is a finite automaton over a finite subset of $A \times B^*$, and $\lambda : F \to B^*$ is a final output function.

Given a word $u = a_1 \cdots a_n \in A^*$ where $a_i \in A$ for all i, a run ρ of $\mathcal{T}$ over u is a sequence $q_0 \cdots q_n$ such that $q_0 \in I$ and $(q_{i-1}, a_i, v_i, q_i) \in \Delta$ for all $i \in [n]$. The *input word* of the run ρ is $u = a_1 \cdots a_n$ and the *output word* of ρ is $v_1 \cdots v_n \cdot \lambda(q_n)$ if q_n is a final state; otherwise, $v_1 \cdots v_n$. As for rational transducers, the relation $R_{\mathcal{T}}$ recognised by $\mathcal{T}$ is defined as the set of pairs (u,v) such that u (resp. v) is the input (resp. output) word of some accepting run. We often confuse $\mathcal{T}$ with $R_{\mathcal{T}}$, and may write $dom(\mathcal{T})$ for $dom(R_{\mathcal{T}})$, or $\mathcal{T}(u)$ for $R_{\mathcal{T}}(u)$.

The *underlying automaton* of $\mathcal{T}$ is the automaton obtained by projection on inputs, i.e. the automaton $\mathcal{A} = (Q, I, \Delta', F)$ such that $\Delta' = \{(q, a, q') \mid \exists (q, a, v, q') \in \Delta\}$. Note that $dom(\mathcal{T}) = L(\mathcal{A})$. The transducer $\mathcal{T}$ is said to be *trim* if its underlying automaton is trim.

The cartesian product, denoted $\mathcal{T}_1 \times \mathcal{T}_2$, of two transducers $\mathcal{T}_i = (Q_i, I_i, \Delta_i, F_i, \lambda_i)$, $i \in [2]$, is the transducer $(Q_1 \times Q_2, I_1 \times I_2, \Delta, F_1 \times F_2, \lambda)$ where $((p_1,p_2), a, (v_1,v_2), (q_1,q_2)) \in \Delta$ if $(p_i, a, v_i, q_i) \in \Delta_i$ for $i \in [2]$, and, $\lambda(p_1,p_2) = (\lambda_1(p_1), \lambda_2(p_2))$ for $(p_1,p_2) \in F_1 \times F_2$.

(Sub)classes of rational functions. Let $\mathcal{T}$ be a real-time transducer. When $R_{\mathcal{T}}$ is functional, $\mathcal{T}$ is said to be functional as well, and the functions recognised by functional transducers are called *rational functions*. If the underlying automaton of $\mathcal{T}$ is unambiguous (i.e., has at most one accepting run on any input), $\mathcal{T}$ is referred to as an *unambiguous transducer*. It is well-known that a function is recognised by real-time transducer iff, it is recognised by a rational transducer [23] iff, it is recognised by an unambiguous transducer [15].

Sequential transducers are those whose underlying automaton is deterministic and they define functions known as *sequential functions*. In that case, we denote the transition function as $\delta : Q \times A \to Q \times B^*$. The functions recognised by transducers that are finite disjoint unions of sequential transducers are called *multi-sequential functions* [14, 24]. The symmetric counterpart of multi-sequential is the class of *series-sequential functions*.

A transducer $\mathcal{T}$ is *series-sequential* if it is a finite disjoint union of sequential transducers $\mathcal{D}_1, \ldots, \mathcal{D}_k$ where additionally, for every $1 \leq i < k$, there is a single transition from a (not necessarily final) state q_i of $\mathcal{D}_i$ to the initial state of the next transducer $\mathcal{D}_{i+1}$. Moreover, the initial state of $\mathcal{T}$ is the initial state of $\mathcal{D}_1$ (the initial states of $\mathcal{D}_i$ is not considered as initial in $\mathcal{T}$, for all $2 \leq i \leq k$). In particular, non-determinism is allowed, for all $1 \leq i < k$, only in state q_i, from which it is possible to move to $\mathcal{D}_{i+1}$ or to stay in $\mathcal{D}_i$. We denote[1] such a transducer as $\mathcal{D}_1 \cdots \mathcal{D}_k$. A function is *series-sequential* if it is recognised by a series-sequential transducer.

Distances between words and word functions. We recall that a metric on a set E is a mapping $d : E^2 \to \mathbb{R}^+ \cup \{\infty\}$ satisfying the separation, symmetry and triangle inequality axioms. Classical metrics between finite words are the *edit distances*. An edit distance between two words is the minimum number of edit operations required to rewrite a word to another if possible, and ∞ otherwise. Depending on the set of allowed edit operations, we get different edit distances. Table 1 gives widely used edit distances with their edit operations.

Table 1 Edit Distances [2].

Edit Distances	Notation	Edit Operations
Hamming	d_h	letter-to-letter substitutions
Longest Common Subsequence	d_{lcs}	insertions and deletions
Levenshtein	d_l	insertions, deletions and substitutions
Damerau-Levenshtein	d_{dl}	insertions, deletions, substitutions and swapping adjacent letters

Distances between words can be lifted to that between functions from words to words.

▶ **Definition 2.1** (Metric over Functions [2]). *Let d be a metric on words over some alphabet B. Given two partial functions $f_1, f_2 : A^* \to B^*$, the distance between f_1 and f_2 is defined as*

$$d(f_1, f_2) = \begin{cases} \sup\{\, d(f_1(w), f_2(w)) \mid w \in dom(f_1)\} & \text{if } dom(f_1) = dom(f_2) \\ \infty & \text{otherwise} \end{cases}$$

It is shown that d is a metric over functions (Proposition 3.2 of [2]). The distance between two functional transducers is defined as the distance between the functions they recognise. A notion closely related to the distance between functions is *diameter* of a relation. The diameter of a relation R w.r.t. a metric d, denoted by $dia_d(R)$, is defined to be the supremum of the distance of every pair in the relation, i.e., $dia_d(R) = \sup\{d(u, v) \mid (u, v) \in R\}$.

The distance between rational functions and the diameter of rational relation w.r.t. the metrics given in Table 1 are computable [2]. The computability of distance and diameter relies on the notion of conjugacy. Two words u and v are conjugate if there exist words x, y such that $u = xy$ and $v = yx$. In other words, they are cyclic shifts of each other. For example, words $aabb$ and $bbaa$ are conjugate with $x = aa$ and $y = bb$. But $aabb$ and $abab$ are not conjugate. Conjugacy is an equivalence relation over words.

▶ **Proposition 2.2.** *Let x, y, x', y', u, v be words and $c, C \in \mathbb{N}$. For any metric d in Table 1, if $d(xu^k y, x'v^{ck}y') \leq C$ for all $k \geq 0$, then $|u| = |v^c|$ and the primitive roots of u and v are conjugate.*

[1] This notation should not be confused with the split-sum operator of [3], which is semantically different.

▶ **Proposition 2.3** ([2]). *Given a rational relation R defined by a transducer $\mathcal{T}$, $dia_d(R) < \infty$ for $d \in \{d_l, d_{lcs}, d_{dl}\}$ if and only if every pair of input-output words generated by loops in $\mathcal{T}$ are conjugate.*

3 Twinning Properties

The class of sequential functions has been characterised by transducers satisfying the so called *twinning property*. In this section, we recall this property and introduce three variants – approximate twinning property, strongly connected twinning property and Hamming twinning property. We also show that these properties on transducers are decidable as well as independent of the representation of the transducers. All these properties are expressed as particular conditions on twinning patterns. A *twinning pattern* for a transducer $\mathcal{T} = (Q, I, \Delta, F, \lambda)$ over A, B is a tuple $(p_1, q_1, p_2, q_2, u, v, u_1, v_1, u_2, v_2) \in Q^4 \times (A^*)^2 \times (B^*)^4$ such that the following runs (graphically depicted) exist:

The longest common prefix of any two words is denoted by $u \wedge v$. The *delay* between u and v, denoted by $delay(u, v)$, is the pair (u', v') such that $u = (u \wedge v)u'$ and $v = (u \wedge v)v'$.

▶ **Definition 3.1** (Twinning Property (**TP**)). *Let $\mathcal{T}$ be a trim transducer. We say that $\mathcal{T}$ satisfies* twinning property *if for each twinning pattern $(p_1, q_1, p_2, q_2, u, v, u_1, v_1, u_2, v_2)$ such that p_1, p_2 are initial, $delay(u_1, u_2) = delay(u_1v_1, u_2v_2)$ holds.*

It is well-known that a function recognised by a transducer $\mathcal{T}$ is sequential iff $\mathcal{T}$ satisfies the twinning property [12, 10]. We now define its approximate variant.

▶ **Definition 3.2** (Approximate Twinning Property (**ATP**)). *A trim transducer $\mathcal{T}$ satisfies* approximate twinning property *if for each twinning pattern $(p_1, q_1, p_2, q_2, u, v, u_1, v_1, u_2, v_2)$ where p_1, p_2 are initial, the words v_1 and v_2 are conjugate.*

▶ **Definition 3.3** (Strongly Connected Twinning Property (**STP**)). *A trim transducer $\mathcal{T}$ satisfies* strongly connected twinning property *if for each twinning pattern $(p_1, q_1, p_2, q_2, u, v, u_1, v_1, u_2, v_2)$ such that $p_1 = p_2$ (not necessarily initial) and p_1, q_1, q_2 are in the same strongly connected component, then $delay(u_1, u_2) = delay(u_1v_1, u_2v_2)$ holds.*

▶ **Definition 3.4** (Hamming Twinning Property (**HTP**)). *A trim transducer $\mathcal{T}$ satisfies* Hamming twinning property *if for each twinning pattern $(p_1, q_1, p_2, q_2, u, v, u_1, v_1, u_2, v_2)$ such that p_1, p_2 are initial, it holds that $|v_1| = |v_2|$ and there is no mismatch between v_1 and v_2, i.e., for all position $i \in [max(|u_1|, |u_2|), min(|u_1v_1|, |u_2v_2|)]$, $(u_1v_1)[i] = (u_2v_2)[i]$.*

▶ **Proposition 3.5.** *Each trim transducer satisfying **TP** also satisfies **ATP**, **STP** and **HTP**.*

The following lemma states that **ATP**, **STP** and **HTP** is preserved between transducers up to finite edit distance, and so in particular between equivalent transducers. This shows that these properties do not depend on the representation of the transductions, not even on the representation of close transductions.

▶ **Lemma 3.6.** *Let $\mathcal{T}$ and $\mathcal{S}$ be two trim transducers satisfying $dom(\mathcal{T}) = dom(\mathcal{S})$, and such that there exists an edit distance d in Table 1 and a constant $C \in \mathbb{N}$ for which*

$$d(v, v') \leq C \text{ for all } u \in dom(\mathcal{T}),\ v \in \mathcal{T}(u),\ v' \in \mathcal{S}(u).$$

Then for every $\mathsf{P} \in \{\boldsymbol{ATP}, \boldsymbol{STP}\}$, $\mathcal{S}$ satisfies P if and only if $\mathcal{T}$ satisfies P. The statement also holds for $d = d_h$ and $\mathsf{P} = \boldsymbol{HTP}$.

Proof sketch. Let us suppose that $\mathcal{S}$ satisfies $\mathsf{P} \in \{\mathbf{ATP}, \mathbf{STP}, \mathbf{HTP}\}$, and show that so does $\mathcal{T}$. The converse is symmetric. We show this for **ATP**, and the proofs for **STP** and **HTP**, following similar arguments, are provided in the full version. Let $(p_1, p_2, q_1, q_2, u, v, u_1, u_2, v_1, v_2)$ be an instance of the twinning pattern for $\mathcal{T}$ with p_1 and q_1 initial. Since $\mathcal{T}$ is trim, there exist words w, w', w_1, w_2 and two accepting states p_f, q_f such that:

$$\begin{array}{llllll} p_1 & \xrightarrow{u|u_1}_{\mathcal{T}} & p_2 & \xrightarrow{v|v_1}_{\mathcal{T}} & p_2 & \xrightarrow{w|w_1}_{\mathcal{T}} & p_f \\ q_1 & \xrightarrow{u|u_2}_{\mathcal{T}} & q_2 & \xrightarrow{v|v_2}_{\mathcal{T}} & q_2 & \xrightarrow{w'|w_2}_{\mathcal{T}} & q_f \end{array}$$

Since $dom(\mathcal{T}) = dom(\mathcal{S})$, for all $i \geq 1$, $uv^iw, uv^iw' \in dom(\mathcal{T}) = dom(\mathcal{S})$. Iterating the loop of $\mathcal{T}$ on v sufficiently many times causes $\mathcal{S}$ to also loop on some power of v. Formally, there exist $a, b, c \in \mathbb{N}$, states $p_1', p_2', q_1', q_2', p_f', q_f'$ and words $u_1', u_2', v_1', v_2', w_1', w_2'$ such that:

$$\begin{array}{llllll} p_1 & \xrightarrow{uv^a|u_1v_1^a}_{\mathcal{T}} & p_2 & \xrightarrow{v^b|v_1^b}_{\mathcal{T}} & p_2 & \xrightarrow{v^cw|v_1^cw_1}_{\mathcal{T}} & p_f \\ q_1 & \xrightarrow{uv^a|u_2v_2^a}_{\mathcal{T}} & q_2 & \xrightarrow{v^b|v_2^b}_{\mathcal{T}} & q_2 & \xrightarrow{v^cw'|v_2^cw_2}_{\mathcal{T}} & q_f \\ p_1' & \xrightarrow{uv^a|u_1'}_{\mathcal{S}} & p_2' & \xrightarrow{v^b|v_1'}_{\mathcal{S}} & p_2' & \xrightarrow{v^cw|w_1'}_{\mathcal{S}} & p_f' \\ q_1' & \xrightarrow{uv^a|u_2'}_{\mathcal{S}} & q_2' & \xrightarrow{v^b|v_2'}_{\mathcal{S}} & q_2' & \xrightarrow{v^cw'|w_2'}_{\mathcal{S}} & q_f' \end{array}$$

Let $C \in \mathbb{N}$ such that $d(\mathcal{T}(x), \mathcal{S}(x)) \leq C$ for all $x \in dom(\mathcal{T})$. Then in particular for all $k \geq 0$:

$$d(u_1v_1^{a+kb+c}w_1, u_1'v_1'^kw_1') \leq C \text{ and } d(u_2v_2^{a+kb+c}w_2, u_2'v_2'^kw_2') \leq C.$$

From the above inequalities, using Proposition 2.2, we get that ρ_{v_i} and $\rho_{v_i'}$, the primitive roots of v_i and v_i' respectively, are conjugate and $|v_i^b| = |v_i'|$ for $i \in [2]$. Since $\mathcal{S}$ satisfies the **ATP**, we also have that v_1' and v_2' are conjugate, and hence their primitive roots are conjugate and $|v_1'| = |v_2'|$. By transitivity of the conjugacy relation, we get that ρ_{v_1} and ρ_{v_2} are conjugate. Further, $|v_1| = |v_2|$ since $|\rho_{v_1}| = |\rho_{v_2}|$ and $|v_1^b| = |v_1'| = |v_2'| = |v_2^b|$. Therefore, v_1 and v_2 are conjugate, and thus the **ATP** holds for $\mathcal{T}$ as well. ◀

Note that the above lemma does not necessarily hold for **TP**. For instance, although $d_l(f_{\mathsf{last}}, \mathsf{id}) < \infty$ and the transducer defining the identity function satisfies **TP**, the transducer defining f_{last} does not.

We now show that checking each of the variants of the twinning property is decidable.

▶ **Lemma 3.7.** *It is decidable whether a transducer satisfies **ATP**, **STP** and **HTP**.*

Proof. Let $\mathcal{T}$ be a transducer that defines a relation R.

Deciding ATP. Let $\mathcal{T}^2$ be the cartesian product of $\mathcal{T}$ by itself. Verifying whether $\mathcal{T}$ satisfies **ATP** reduces to checking that every loop in $\mathcal{T}^2$ produces a pair of conjugate output words. For each state $(p, q) \in \mathcal{T}^2$, we define a rational relation $R_{(p,q)}$ consisting of pairs of output words produced by the loops in $\mathcal{T}^2$ rooted at (p, q). The transducer defining $R_{(p,q)}$ is obtained from $\mathcal{T}^2$ by disregarding the inputs and setting both the initial and final state to be (p, q). It is known that we can decide whether every pair of words in a given rational relation is conjugate [1], which entails decidability of **ATP**.

Deciding STP. The twinning property is decidable in polynomial time [10]. To decide **STP**, we first decompose the transducer into SCCs (in polynomial time). Then, for each SCC and each state p, the SCC is seen as a transducer with initial state p on which we check **TP**.

Deciding HTP. We prove that it can be decided in PTime whether a transducer does not satisfy **HTP**. The **HTP** can be decomposed as a conjunction of two properties : $\mathbf{HTP}_{lgth}$ on lengths and $\mathbf{HTP}_{mism}$ on mismatches. The (negation of) $\mathbf{HTP}_{lgth}$ can be directly expressed in the pattern logic of [18], whose model-checking is in PTime. Then, we reduce the problem of deciding the $\mathbf{HTP}_{mism}$ to the emptiness problem of a Parikh automaton of polynomial size, in the size of the transducer $\mathcal{T}$. We recall that a Parikh automaton of dimension d is an NFA extended with vectors in $\mathbb{Z}^d$ on its transitions. It accepts a word if there exists a run of the NFA that reaches an accepting state, and the sum of the vectors seen along the run belongs to some semi-linear set $W \subseteq \mathbb{Z}^d$, given for example as a Presburger formula $\phi(x_1, \dots, x_d)$. The emptiness problem is known to be decidable in PTime when both d and ϕ are constant [16, 18]. Our reduction follows standard ideas, and falls in this particular case. Let us give a bit more details.

A twinning pattern t can be encoded as a word u_t over $\Delta^2 \cup \{\#\}$, where Δ is the transition relation of $\mathcal{T}$. In this construction, a run of $\mathcal{T}$ is seen as a sequence of transitions. So, a twinning pattern consists of four runs, r_1, r'_1 and r_2, r'_2 where for all $i = 1, 2$, r_i is the run such that $p_i \xrightarrow{u|u_i} q_i$ and r'_i is the run such that $q_i \xrightarrow{v|v_i} q_i$. The four runs are encoded as a word $u_t = (r_1 \otimes r_2)\#(r'_1 \otimes r'_2)$ where $r_1 \otimes r_2$ is the convolution of r_1 and r_2, i.e. the overlapping of r_1 and r_2 (and similarly for $r'_1 \otimes r'_2$). It should be clear that the set of words u_t for all twinning pattern t such that p_1 and p_2 are initial states is a regular language, recognizable by an NFA of polysize in the size of $\mathcal{T}$.

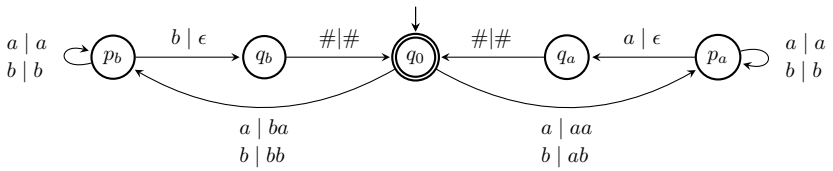
In order to check the condition that there is a mismatch between u_1v_1 and u_2v_2 on a position common to v_1 and v_2, the NFA is extended with two counters c_1 and c_2, and the linear acceptance condition $c_1 = c_2 = 0$ (making it a Parikh automaton). Those two counters are used to guess the mismatching position. Initially, using an ϵ-loop, those two counters are incremented in parallel. At the end of this first phase, they therefore hold the same value, say $i \in \mathbb{N}$. Then, the Parikh automaton is built in such a way that it checks that $|u_1| < i \leq |u_1v_1|$ and $|u_2| < i \leq |u_2v_2|$, and $(u_1v_1)[i] \neq (u_2v_2)[i]$. To do so, while reading any transition of r_j producing some word α_j, $j = 1, 2$, c_j is decremented by $|\alpha_j|$. The same is done when reading transitions of r'_j, but the automaton can non-deterministically guess that the counter c_j is equal to 0, and store (in its state) the corresponding letter in α_j. From then on, it never decrements the counter c_j again. When the whole input $(r_1 \otimes r_2)\#(r'_1 \otimes r'_2)$ has been read, the automaton has two letters stored in its state. It accepts the input, if those two letters are different, and if $c_1 = c_2 = 0$. ◀

4 Approximate determinisation of rational functions

In this section, we define *approximately determinisable functions* and give decidable properties on transducers to characterise them.

▶ **Definition 4.1** (Approximate determinisation). *A rational function $f : A^* \to B^*$ is* approximately determinisable *w.r.t. a metric d if there exists a sequential function $g : A^* \to B^*$ such that $d(f, g) < \infty$. In this case, we also say that g* approximately determinises *f w.r.t. d.*

▶ **Example 4.2.** The function f_{last} from the introduction is approximately determinisable w.r.t. Levenshtein, while not w.r.t. Hamming distance. The function f^*_{last} (depicted below) that maps $u_1\# \cdots u_n\#$, for $n \geq 1$, to $f_{\mathsf{last}}(u_1)\# \cdots f_{\mathsf{last}}(u_n)\#$ for some separator $\#$ is approximately determinisable w.r.t neither Levenshtein nor Hamming distance.



The approximate determinisation problem asks whether a rational function given as a functional transducer is approximately determinisable. We prove the following.

▶ **Theorem 4.3.** *The approximate determinisation problem for rational functions w.r.t. Levenshtein family (d_l, d_{lcs}, d_{dl}) and Hamming (d_h) distance are decidable.*

To prove the theorem, we show that **ATP** and **STP** characterise rational functions that can be approximately determinised w.r.t Levenshtein family (Lemma 4.8). Similarly, we establish that **HTP** and **STP** characterise rational functions that can be approximately determinised w.r.t Hamming distance. Theorem 4.3 then follows directly, as these three properties are decidable (Lemma 3.7). We now outline the proof strategy. The full proof for Levenshtein family is presented in Section 4.1, while the proof for Hamming distance, which follows a similar approach, is provided in the full version.

Levenshtein family. We show with Proposition 4.4 that **ATP** and **STP** are necessary conditions for approximate determinisation with respect to Levenshtein family, a consequence of Lemma 3.6. The main challenge lies in proving that these properties are sufficient. To prove it, we first show that **ATP** alone suffices for certain subclasses of functional transducers: it enables the approximate determinisation of multi-sequential (Lemma 4.6) and unambiguous series-sequential (Lemma 4.7) functions. However, for rational functions in general, **ATP** does not suffice. For example, the transducer above for f^*_{last} satisfies **ATP** but is not approximately determinisable. To extend this result to all rational functions, we incorporate **STP**. Given a functional transducer $\mathcal{T}$ satisfying **STP**, we transform each strongly connected component of $\mathcal{T}$ into a sequential transducer, effectively decomposing $\mathcal{T}$ into a finite union of concatenations of sequential transducers. We then leverage our results for series-sequential and multi-sequential functions to approximate this structure with a sequential function (Lemma 4.8).

Figure 1 illustrates the main construction technique used in these proofs: Starting with a transducer $\mathcal{T}$ that we aim to approximate, we construct a sequential transducer $\mathcal{D}_1$ as follows. We apply the powerset construction to $\mathcal{T}$, introducing a distinguished state (marked by a • in the figure) in each subset. The output is determined by the distinguished state's production. If the distinguished state reaches a point where it has no continuation, we simply transition to another distinguished state. We show that **ATP**, combined with a carefully chosen priority scheme for selecting distinguished states, ensures bounded Levenshtein distance.

Hamming Distance. The proof strategy is similar to the Levenshtein setting: We first show that **HTP** and **STP** are necessary conditions for approximate determinisation with respect to Hamming distance, we show that **HTP** alone suffices for the approximate determinisation of multi-sequential and series-sequential functions, and then we conclude by using **STP** to transform functional transducers into a finite unions of concatenations of sequential transducers.

While the core ideas remain similar to those used for the Levenshtein family, the constructions required for the Hamming distance, illustrated in Figure 1, are more intricate. Approximating the transducer $\mathcal{T}$ with respect to the Levenshtein distance (as shown by $\mathcal{D}_1$

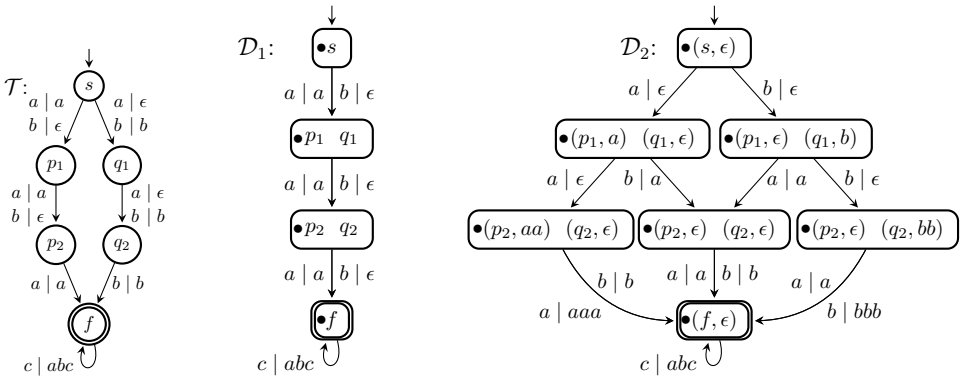


Figure 1 An unambiguous non-deterministic transducer $\mathcal{T}$, along with two sequential approximations $\mathcal{D}_1$ and $\mathcal{D}_2$ with respect to the Levenshtein distance, respectively Hamming distance.

in the figure) allows us, at each step, to select a run, produce its output, and disregard other possible runs. However, for the Hamming distance, it is crucial to carefully track the length difference between the produced output and the potential outputs of alternative runs. For instance, compare the outputs of $\mathcal{T}$, $\mathcal{D}_1$, and $\mathcal{D}_2$ after reading *babcccc* :

$$\begin{aligned} \mathcal{T}(\texttt{babcccc}) &= \texttt{bbabcabcabcabc}, \\ \mathcal{D}_1(\texttt{babcccc}) &= \texttt{aabcabcabcabc}, \\ \mathcal{D}_2(\texttt{babcccc}) &= \texttt{ababcabcabcabc}. \end{aligned}$$

We observe that after reading the input *bab*, $\mathcal{D}_1$ realizes that its distinguished state is incorrect and jumps to another state. However, this shift causes a misalignment with $\mathcal{T}$, and reading additional c's results in arbitrarily many mismatches.[2] In contrast, $\mathcal{D}_2$ keeps in memory the delay relative to other runs. Although it may still introduce mismatches along the way, it ensures that when the distinguished run terminates, it adjusts the output while transitioning to another run, preventing long-term misalignment with $\mathcal{T}$.

4.1 Approximate determinisation for Levenshtein Family

We give a decidable characterisation of approximately determinisable rational functions w.r.t. Levenshtein family of distances – Levenshtein (d_l), Longest common subsequence (d_{lcs}) and Damerau-Levenshtein (d_{dl}). They are all equivalent up to boundedness (Lemma 2.1 and Remark 7 of [2]), i.e., for any two rational functions f, g, $d_{lcs}(f, g) < \infty \iff d_l(f, g) < \infty \iff d_{dl}(f, g) < \infty$. We show that a rational function is approximately determinisable w.r.t. Levenshtein family if and only if the transducer that defines the function satisfies both **ATP** and **STP**. One direction is a consequence of Lemma 3.6 as follows.

▶ **Proposition 4.4.** *If a rational function given by a trim transducer $\mathcal{T}$ is approximately determinisable w.r.t. a metric $d \in \{d_l, d_{lcs}, d_{dl}\}$ then $\mathcal{T}$ satisfies both **ATP** and **STP**.*

Proof. Given that the rational function given by $\mathcal{T}$ is approximately determinisable, i.e., there exists a sequential function given by a deterministic transducer $\mathcal{D}$ such that $d(\mathcal{T}, \mathcal{D}) < \infty$. Since $\mathcal{D}$ is a sequential transducer, it satisfies the twinning property, and consequently, $\mathcal{D}$ also satisfies **ATP** and **STP** by Proposition 3.5. Since $d(\mathcal{T}, \mathcal{D}) < \infty$, we can conclude that $\mathcal{T}$ also satisfies **ATP** as well as **STP** by applying Lemma 3.6. ◀

[2] The Levenshtein distance remains bounded, as inserting a letter at the start resynchronizes both outputs.

Towards proving the other direction, we prove the following lemma, which provides a bound on the distance between the output words produced by distinct runs of a transducer on the same prefix of an input word using Proposition 2.3.

▶ **Lemma 4.5.** *Let $\mathcal{T}$ be a trim transducer satisfying the **ATP**. Then, there exists a constant $N_{\mathcal{T}} \in \mathbb{N}$ such that for any two output words $v, v' \in B^*$ produced via two distinct runs of $\mathcal{T}$ from an initial state on the same prefix of an input word, $d(v, v') \leq N_{\mathcal{T}}$ for $d \in \{d_l, d_{lcs}, d_{dl}\}$.*

Proof. Let $\mathcal{T}^2$ be the cartesian product of $\mathcal{T}$ by itself. By designating all states of $\mathcal{T}^2$ as final, and disregarding the input word, we obtain a new transducer that defines the relation R_o, consisting of all pairs of output words produced by distinct runs of T on the same prefix of an input word. Since $\mathcal{T}$ satisfies **ATP**, every pair of output words produced by loops in $\mathcal{T}^2$ on any input are conjugate. Consequently, since R_o is obtained by ignoring the input word from $\mathcal{T}^2$, it satisfies the condition in Proposition 2.3. Hence, the diameter of R_o w.r.t. Levenshtein family of distances is bounded.

Let $dia_d(R_o) \leq k$. By the definition of diameter, it follows that $d(u, v) \leq k$ for all $(u, v) \in R_o$. As a result, the distance between any two output words produced by distinct runs of $\mathcal{T}$ on the same word is at most k. Setting $N_T = k$ completes the proof. ◀

For subclasses of rational functions, namely multi-sequential and series-sequential functions, we show that **ATP** is a sufficient condition for approximate determinisation.

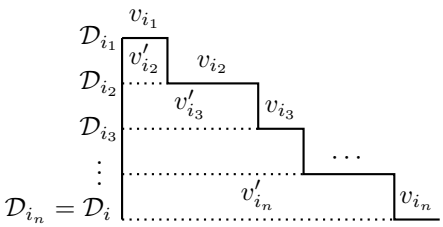
▶ **Lemma 4.6.** *A multi-sequential function given by a trim transducer $\mathcal{T}$ is approximately determinisable w.r.t. a metric $d \in \{d_l, d_{lcs}, d_{dl}\}$ iff $\mathcal{T}$ satisfies the **ATP**.*

Proof. ($\rightarrow$) is direct by Proposition 4.4, and we show ($\leftarrow$). Since the transducer $\mathcal{T}$ is multi-sequential, it is equivalent to some finite union of sequential transducers $\mathcal{U} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_k$ for some $k \in \mathbb{N}$ where $\mathcal{D}_i = (Q_i, s_i, \delta_i, F_i, \lambda_i)$ is a sequential trim transducer for $i \in [k]$. By Lemma 3.6 and since $\mathcal{T}$ satisfies **ATP**, the transducer $\mathcal{U}$ also satisfies **ATP**. We construct a sequential transducer $\mathcal{D}$ that approximately determinises $\mathcal{T}$, which intuitively is simply the cartesian product of $\mathcal{D}_1, \ldots, \mathcal{D}_k$ which on each transition produces the output of the smallest index transducer $\mathcal{D}_i$ for which that transition is defined. Let $\mathcal{D} = (Q, s, \delta, F, \lambda)$ where

1. The set of states $Q = Q'_1 \times Q'_2 \times \cdots \times Q'_k$ is the cartesian product of the state set Q'_i for $i \in [k]$ such that $Q'_i = Q_i \cup \{d\}$ where d represents a dead state. The initial state is $s = (s_1, s_2, \cdots, s_k)$, and the set of final states F is $\{(p_1, p_2, \ldots, p_k) \mid \exists i\ p_i \in F_i\}$.
2. The function $\delta : Q \times A \to Q \times B^*$ is defined as: $\delta((p_1, p_2, \ldots, p_k), a) = ((q_1, q_2, \ldots, q_k), x)$ where for each $i \in [k]$, either $\delta_i(p_i, a) = (q_i, x_i)$ or, if $p_i = d$ or $\delta_i(p_i, a)$ is undefined, then $q_i = d$. The output x is set to x_j, where $j \in [k]$ is the smallest index for which the transition $\delta_j(p_j, a)$ is defined.
3. The output function $\lambda : F \to B^*$ is defined as $\lambda((p_1, p_2, \ldots, p_k)) = \lambda_i(p_i)$ where $i \in [k]$ is the smallest index such that $p_i \in F_i$.

We show that $d(\mathcal{D}, \mathcal{T}) < \infty$ w.r.t. Levenshtein family. From the construction, it is clear that $\mathcal{D}$ and $\mathcal{T}$ have the same domain. Consider an input word $u \in dom(\mathcal{T})$. Let $i \in [k]$ be the smallest index such that $u \in dom(\mathcal{D}_i)$, with output word, say v. The transducer $\mathcal{D}$ on input u produces output, say v', by concatenating output on each transition over input word produced by the smallest index transducer.

Thus, v' can be decomposed into $v_{i_1} v_{i_2} \cdots v_{i_n}$ where $i_1 < i_2 < \cdots < i_n = i$ and for each $i_j \in [i]$, v_{i_j} is the output produced by $\mathcal{D}_{i_j}$ along the partial run of u. For each v_{i_j}, let v'_{i_j} denote the prefix of the output produced by $\mathcal{D}_{i_j}$ upto v_{i_j}. The output produced by $\mathcal{T}$ on u via $\mathcal{D}_i$ is $v = v'_i v_i = v'_{i_n} v_{i_n}$.



Observe that $v'_{i_j} v_{i_j}$ and v'_{i_j+1} (for $i_1 < i_j < i_n$) is the output produced by $\mathcal{D}_{i_j}$ and $\mathcal{D}_{i_j+1}$ on the same prefix of input u. By Lemma 4.5, we obtain $d(v'_{i_j} v_{i_j}, v'_{i_j+1}) \leq N_T$. Similarly, since v_{i_1}, v'_{i_2} are the outputs of $\mathcal{D}_{i_1}$ and $\mathcal{D}_{i_2}$ on the same input prefix, we get $d(v_{i_1}, v'_{i_2}) \leq N_T$.

$$\begin{aligned} d(v_{i_1} v_{i_2} v_{i_3} \dots v_{i_n}, v) &= d(v_{i_1} v_{i_2} v_{i_3} \cdots v_{i_n}, v'_{i_n} v_{i_n}) \text{ (Since } v = v'_{i_n} v_{i_n}) \\ &\leq d(v_{i_1} v_{i_2} v_{i_3} \cdots v_{i_n}, v'_{i_2} v_{i_2} v_{i_3} \cdots v_{i_n}) + d(v'_{i_2} v_{i_2} v_{i_3} \cdots v_{i_n}, v'_{i_3} v_{i_3} \cdots v_{i_n}) \\ &\quad + \cdots + d(v'_{i_n-1} v_{i_n-1} v_{i_n}, v'_{i_n} v_{i_n}) \text{ (Applying triangle inequality of } d) \\ &\leq n \cdot N_T \text{ (using Lemma 4.5)} \end{aligned}$$

In fact, on any input word, the distances between the outputs produced by $\mathcal{D}$ and $\mathcal{T}$ is less than or equal to $k \cdot N_T$, as there can be at most k switches between runs. Hence, we get that $d(\mathcal{D}, \mathcal{T})$ is bounded. ◀

The characterisation of Lemma 4.6 also holds for unambiguous series-sequential functions.

▶ **Lemma 4.7.** *Let $\mathcal{T} = \mathcal{D}_1 \cdots \mathcal{D}_k$ be an unambiguous transducer where each $\mathcal{D}_i$ is a sequential trim transducer for $i \in [k], k > 1$. The series-sequential function defined by $\mathcal{T}$ is approximately determinisable w.r.t. a metric $d \in \{d_l, d_{lcs}, d_{dl}\}$ if and only if $\mathcal{T}$ satisfies **ATP**.*

Proof. ($\rightarrow$) follows from Proposition 4.4, and we prove ($\leftarrow$). Similarly to the proof in Lemma 4.6, we construct a sequential transducer $\mathcal{D}$ using a subset construction of $\mathcal{T}$ such that $\mathcal{D}$ approximately determinises $\mathcal{T}$. Although $\mathcal{T}$ is functional, and each $\mathcal{D}_i$, $i \in [k]$, is sequential, the non-determinism arises between the transitions between one $\mathcal{D}_i$ to the other. We assume an ordering for the states of $\mathcal{T}$ for each D_i. Intuitively, $\mathcal{D}$ stores a subset of states of $\mathcal{T}$ to capture all possible *active runs* on any input word. It produces the output of the active run of the smallest indexed sequential transducer, called the *producing run*. Upon termination of the producing run, it switches to the next active run of the smallest indexed sequential transducer. Since $\mathcal{T}$ is unambiguous, there is at most one accepting run on any input. Formally, $\mathcal{D} = (Q, s_0, \delta, F, \lambda)$ where

1. Q is the power set of the states in $\mathcal{T}$. Each set $S \in Q$ has exactly one state marked with a $\bullet$ to denote the currently producing run.
2. s_0 is the initial state of $\mathcal{T}$ and is marked with a $\bullet$.
3. The transition function $\delta : Q \times A \to Q \times B^*$ is defined as follows: $\delta(P, a) = (S, x)$ where S consists of all states q in $\mathcal{T}$ reachable from a state $p \in P$ via transition (p, a, q, x_{pq}) in $\mathcal{T}$. The output word x is set as follows. Let p be the $\bullet$ marked state in P that belongs to the sequential transducer $\mathcal{D}_i$ for some $i \in [k]$.
   a. If a transition (p, a, q, x_{pq}) in $\mathcal{T}$ exists within the same sequential transducer $\mathcal{D}_i$, then $x = x_{pq}$ and q is $\bullet$ marked in S.
   b. If no transition exists from p on a within $\mathcal{D}_i$, but a transition (p, a, q, x_{pq}) in $\mathcal{T}$ exists to a different sequential transducer, i.e., q belongs to $\mathcal{D}_{i+1}$, then $x = x_{pq}$ and q is $\bullet$ marked in S. Such a transition is called a *switch* (to $\mathcal{D}_{i+1}$).

   c. Otherwise, choose the smallest numbered state $p' \in P$ that belongs to the smallest indexed sequential transducer $\mathcal{D}_j$ where $j \in [k]$ with a defined transition $(p', a, q', x_{p'q'})$ in $\mathcal{T}$, and set $x = x_{p'q'}$ and $\bullet$ mark q' in S. Such a transition is also called a switch (to $\mathcal{D}_j$).

4. F is the set of all states $P \in Q$ such that P contains a final state of $\mathcal{T}$.
5. The output function $\lambda : F \to B^*$ is defined, for $P \in F$, as $\lambda(P) = \lambda_{\mathcal{T}}(p)$ for some arbitrary final state $p \in P$ of $\mathcal{T}$, where $\lambda_{\mathcal{T}}$ is the output function of $\mathcal{T}$.

The number of states of $\mathcal{D}$ is exponential in the number of states of $\mathcal{T}$. We now argue that the number of switches in the run of $\mathcal{D}$ between the active runs of $\mathcal{T}$ on any input word is less than the number of states in $\mathcal{T}$, and hence finite. Let n_i be the number of states in $\mathcal{D}_i$ for $i \in [k]$. Consider the run of $\mathcal{D}$ on an arbitrary input word. Initially, only the initial state of $\mathcal{D}_1$ is active in $\mathcal{D}$. As the run proceeds, if P is the set of states of $\mathcal{T}$ reached so far, then by construction of $\mathcal{D}$, the only $\bullet$ marked state in P is the last state of the active run of $\mathcal{T}$ that belongs to the smallest indexed $\mathcal{D}_i$. If this run eventually dies, then two cases can happen: (1) $\mathcal{D}$ switches to another active run in $\mathcal{D}_i$, or, (2) if none exists, $\mathcal{D}$ switches to some active run in $\mathcal{D}_j$, where $j > i$ is minimal. If case (2) happens, then no more states of $\mathcal{D}_i$ are active, so $\mathcal{D}$ will never switch again to $\mathcal{D}_i$ in the future. The number of times case (1) can happen is bounded by n_i. Indeed, at most n_i states of $\mathcal{D}_i$ can be active at any moment, and since $\mathcal{D}_i$ is sequential, the number of active states in $\mathcal{D}_i$ can only decrease.

Now, if $\mathcal{D}$ eventually switches to $\mathcal{D}_j$, at most n_j states in $\mathcal{D}_j$ are active, so at most n_j switches of type (1) can happen in $\mathcal{D}_j$, and so on until $\mathcal{D}$ eventually terminates in some sequential transducer $\mathcal{D}_l$ for $l \geq j$. Therefore, in the worst case, $\mathcal{D}$ switches n_i times in $\mathcal{D}_i$ before switching to $\mathcal{D}_{i+1}$ for all $i \in \{1, \ldots, k-1\}$. So, the overall number of switches in the run of $\mathcal{D}$ is at most $N = \sum_{i=1}^{k} n_i$, which is exactly the number of states in $\mathcal{T}$.

Now we show that $d(\mathcal{D}, \mathcal{T}) < \infty$ w.r.t. Levenshtein family. From the construction, it is clear that $\mathcal{D}$ and $\mathcal{T}$ have the same domain.

Consider an input word u accepted by $\mathcal{T}$. Since $\mathcal{T}$ is unambiguous, there is exactly one run in $\mathcal{T}$ that accepts u. Let $v = \mathcal{T}(u)$ and $v' = \mathcal{D}(u)$. From the construction of $\mathcal{D}$, v' can be decomposed into $v_{s_1} v_{s_2} \cdots v_{s_n}$ accommodating $n \in [N]$ switches between the active runs of $\mathcal{T}$ on the prefixes of u, where each v_{s_i}, $i \in [n]$ is the output produced by an active run r_i on the prefix of u. For each v_{s_i}, let v'_{s_i} denote the prefix of the output produced by the run r_i up to v_{s_i}.

Observe that $v'_{s_i} v_{s_i}$ and $v'_{s_{i+1}}$ (for $1 < i < n$) is the output produced by $\mathcal{D}_1 \cdots \mathcal{D}_{j_i}$ and $\mathcal{D}_1 \cdots \mathcal{D}_{j_{i+1}}$ on the same prefix of u where $j_i, j_{i+1} \in [k]$. By using Lemma 4.5, we obtain $d(v'_{s_i} v_{s_i}, v'_{s_{i+1}}) \leq N_T$. Similarly, since v_{s_1}, v'_{s_2} are the outputs of $\mathcal{D}_1 \cdots \mathcal{D}_{j_1}$ and $\mathcal{D}_1 \cdots \mathcal{D}_{j_2}$ on the same input prefix, it follows that $d(v_{s_1}, v'_{s_2}) \leq N_T$. As a consequence,

$$
\begin{aligned}
&d(v_{s_1} v_{s_2} \ldots v_{s_n}, v) \\
&\qquad = d(v_{s_1} v_{s_2} \cdots v_{s_n}, v'_{s_n} v_{s_n}) \ (v = v'_{s_n} v_{s_n} \text{ since } r_{s_n} \text{ is the only accepting run of } u) \\
&\qquad \leq d(v_{s_1} v_{s_2} \cdots v_{s_n}, v'_{s_2} v_{s_2} \cdots v_{s_n}) + d(v'_{s_2} v_{s_2} \cdots v_{s_n}, v'_{s_3} v_{s_3} \cdots v_{s_n}) \\
&\qquad\quad + \cdots + d(v'_{s_{n-1}} v_{s_{n-1}} v_{s_n}, v'_{s_n} v_{s_n}) \text{ (Applying triangle inequality of } d) \\
&\qquad \leq n \cdot N_{\mathcal{T}} \text{ (using Lemma 4.5)}
\end{aligned}
$$

Therefore, $d(\mathcal{D}(u), \mathcal{T}(u)) \leq n \cdot N_T \leq N \cdot N_{\mathcal{T}}$. This holds for any $u \in dom(\mathcal{T})$, and N being the number of states in $\mathcal{T}$, we get that $d(\mathcal{D}, \mathcal{T})$ is bounded. ◀

We extend the characterisation to rational functions, where **STP** is also required to decompose the function into a finite union of series-sequential functions, which can then be transformed into a multi-sequential function using the properties of **ATP**.

▶ **Lemma 4.8.** *A rational function given by a trim transducer $\mathcal{T}$ is approximately determinisable w.r.t. a metric $d \in \{d_l, d_{lcs}, d_{dl}\}$ iff $\mathcal{T}$ satisfies both **ATP** and **STP**.*

Proof. ($\rightarrow$) follows from Proposition 4.4. We prove ($\leftarrow$). Assume that the rational function is given by an unambiguous transducer $\mathcal{T}$ with set of states Q. Disregarding the labels on transitions, decompose $\mathcal{T}$ into maximal SCCs $S_1, \ldots, S_k \subseteq Q$. Consider the set of paths Π of the form $\pi = S_{i_1} t_{i_1} S_{i_2} \ldots t_{i_{n-1}} S_{i_n}$ such that S_{i_1} is an SCC which contains an initial state, S_{i_n} is an SCC which contains a final state, and for all $1 \leq k < n$, t_{i_k} is a transition of $\mathcal{T}$ from a state of S_{i_k} to some state of $S_{i_{k+1}}$. Let $\mathcal{T}_\pi$ denote the trim subtransducer of $\mathcal{T}$ that removes all the transitions in $\mathcal{T}$ except the transitions t_{i_k} $(1 \leq k < n)$ and the transitions occurring in the SCCs S_{i_k} $(1 \leq k \leq n)$.

Note that since the SCCs are maximal, the set Π is finite. Now, it is straightforward to see that $\mathcal{T} \equiv \mathcal{U} = \bigcup_{\pi \in \Pi} \mathcal{T}_\pi$. From Lemma 3.6, we deduce $\mathcal{U}$ satisfies both **ATP** as well as **STP**. Moreover, given $\mathcal{T}$'s unambiguity, each input accepted by $\mathcal{T}$ is accepted by exactly one $\mathcal{T}_\pi$ and, each SCC within a $\mathcal{T}_\pi$ has a single entry and exit point.

Since $\mathcal{U}$ satisfies **STP**, each SCC in T_π satisfies **TP**, with initial state being the unique entry point of the SCC. Hence we can determinise each SCC in $\mathcal{T}_\pi$ and can obtain a series-sequential transducer that is equivalent to $\mathcal{T}_\pi$ and indeed satisfies **ATP** by Lemma 3.6. From Lemma 4.7, there exists a sequential transducer $\mathcal{D}_\pi$ for each $\mathcal{T}_\pi$, such that $d(\mathcal{D}_\pi, \mathcal{T}_\pi)$ is finite.

Let $d(\mathcal{T}_\pi, \mathcal{D}_\pi) \leq k_\pi$ for some $k_\pi \in \mathbb{N}$. Consequently, the distance between $\mathcal{T}$ and the new transducer $\mathcal{U}' = \bigcup_{\pi \in \Pi} \mathcal{D}_\pi$ is bounded where $d(\mathcal{T}, \mathcal{U}') = d(\mathcal{U}, \mathcal{U}') \leq \max\{k_\pi \mid \pi \in \Pi\}$. Further, since $\mathcal{T}$ satisfies **ATP**, we deduce $\mathcal{U}'$ also satisfies **ATP** by Lemma 3.6. Being multi-sequential and satisfying **ATP**, $\mathcal{U}'$ can be further approximately determinised using the construction outlined in Lemma 4.6. Let $\mathcal{D}$ be the sequential transducer such that $d(\mathcal{U}', \mathcal{D}) < \infty$. Since d is a metric, $d(\mathcal{T}, \mathcal{D}) \leq d(\mathcal{T}, \mathcal{U}') + d(\mathcal{U}', \mathcal{D})$. Since both $d(\mathcal{T}, \mathcal{U}')$ and $d(\mathcal{U}', \mathcal{D})$ are finite, $d(\mathcal{T}, \mathcal{D})$ is also finite. ◀

5 Approximate decision problems for rational relations

In this section, we consider three possible generalisations of the approximate determinisation problem to rational relations. We describe those generalisations informally. The first one asks to decide whether a rational relation is close to some rational function. We call it the approximate functionality problem. The second one, that we still call determinisation problem, amounts to decide, given a rational relation R, whether it is *almost* a sequential function. The third generalisation we consider is an approximate uniformisation problem, which asks, given R, whether there exists a sequential function s which is close to a function f, whose graph is included in R. We however show that this problem is undecidable.

We now proceed with the formal definitions and statements of our results. First, we need to extend the notion of distance from functions of words to binary relations of words. Towards this, we use Hausdorff distance between languages, defined as

$$d_H(L, L') = \max\left\{ \sup_{w \in L} \inf_{w' \in L'} d(w, w'), \sup_{w' \in L'} \inf_{w \in L} d(w, w') \right\}.$$

Given a metric d on words, and two relations $R_1, R_2 \subseteq A^* \times B^*$, the distance between R_1 and R_2 is defined as follows.

$$d(R_1, R_2) = \begin{cases} \sup\{ d_H(R_1(w), R_2(w)) \mid w \in dom(R_1) \} & \text{if } dom(R_1) = dom(R_2) \\ \infty & \text{otherwise} \end{cases}$$

Therefore, $d(R_1, R_2) < \infty$ iff $dom(R_1) = dom(R_2)$ and there exists $k \in \mathbb{N}$ such that for all word u in the domain and any output v_1 of R_1 on u, there exists some output v_2 of R_2 on u, such that $d(v_1, v_2) < k$, and symmetrically. In fact, d is a metric on relations (see full version).

Approximate functionality problem.

▶ **Definition 5.1** (Approximate Functionalisation). *A rational relation R is* approximately functionalisable *w.r.t. a metric d if there exists a rational function f such that $d(R, f) < \infty$.*

The approximate functionality problem asks, given R represented by a transducer, whether it is approximately functionalisable w.r.t. d. Towards this, we define the following value for a relation R and metric d, which measures how different are output words over the same input. More precisely, it is the maximal distance between any two output words over the same input word, by R:

$$\mathsf{diff}_d(R) = \sup_{u \in dom(R)} \sup_{v_1, v_2 \in R(u)} d(v_1, v_2)$$

▶ **Lemma 5.2.** *For a rational relation R given as a rational transducer, $\mathsf{diff}_d(R)$ is computable for all metrics given in Table 1.*

Proof. Given a rational relation R, we can construct a rational relation R_o that consists of all pairs of output words of R on any input, i.e., $R_o = \{(v_1, v_2) \mid \exists u \in dom(R), (u, v_1), (u, v_2) \in R\}$. If R is a relation defined by a transducer $\mathcal{T}$, then the transducer obtained by $\mathcal{T} \times \mathcal{T}$ (cartesian product of $\mathcal{T}$ by itself) by ignoring the input word is a transducer that defines the relation R_o. Observe that $\mathsf{diff}_d(R)$ is equivalent to the diameter of R_o w.r.t. d. It is shown in [2] that diameter of a rational relation is computable for all metrics given in Table 1. Hence, for those metrics, $\mathsf{diff}_d(R)$ is computable. ◀

We now characterise rational relations which are approximately functionalisable.

▶ **Lemma 5.3.** *A rational relation R is approximately functionalisable w.r.t. a metric d if and only if $\mathsf{diff}_d(R) < \infty$.*

Proof. ($\rightarrow$) Assume that R is approximately funtionalisable, i.e., there exists a rational function f such that $d(R, f) < \infty$. Therefore, $dom(R) = dom(f)$ and there exists an integer k such that for each input word $u \in dom(R)$, for each output word $v \in R(u)$, $d(v, f(u)) \leq k$ (since f is a function). By triangle inequality of metric d, the distance between any two arbitrary output words $v_1, v_2 \in R(u)$ on any input $u \in dom(R)$ is $d(v_1, v_2) \leq d(v_1, f(u)) + d(f(u), v_2) \leq 2k$. Since this holds for any input in the domain of R, we get $\mathsf{diff}_d(R) = \sup_{u \in dom(R)} \sup_{v_1, v_2 \in R(u)} d(v_1, v_2) \leq 2k$.

($\leftarrow$) Assume that $\mathsf{diff}_d(R) < \infty$. Let $\mathsf{diff}_d(R) \leq k$ for some $k \in \mathbb{N}$. We show that any uniformiser of the relation (a function of same domain as the relation whose graph is included in the relation) is a function that approximately functionalises the relation. Let f be a uniformiser of R. We prove that $d(R, f) < \infty$. Since f is a uniformiser of R, $dom(R) = dom(f)$, and for all $u \in dom(R)$, it holds that $(u, f(u)) \in R$. Since $\mathsf{diff}_d(R) \leq k$, the distance between the outputs of R on any input is less than or equal to k. Thus, for any input word $u \in dom(R)$, for each output word $v \in R(u)$, $d(v, f(u)) \leq k$ (since $f(u)$ is also an output of $R(u)$). Hence, $d(R, f) < \infty$, i.e., R is approximately functionalisable w.r.t. d. ◀

Since $\mathsf{diff}_d(R)$ is computable (see Lemma 5.2), we get the following result.

▶ **Theorem 5.4.** *The approximate functionality problem for rational relations given as rational transducers w.r.t. a metric given in Table 1 is decidable.*

Approximate determinisation problem. A rational relation R is said to be *approximately determinisable* for a metric d if it is almost a sequential function with respect to d. Formally, it means that there exists a sequential function f such that $d(R, f) < \infty$. We show that the associated decision problem, that we call approximate determinisation problem, is decidable for Levenshtein family of distances. In fact, the characterisation for approximate determinisation of rational functions also holds for rational relations.

▶ **Lemma 5.5.** *A rational relation defined by a trim transducer $\mathcal{T}$ is approximate determinisable w.r.t. Levenshtein family (d_l, d_{lcs}, d_{dl}) if and only if $\mathcal{T}$ satisfies **ATP** and **STP**.*

Proof. Let R be a rational relation given by a transducer $\mathcal{T}$, and let $d \in \{d_l, d_{lcs}, d_{dl}\}$. The proof of direction $(\rightarrow)$ is the same as Proposition 4.4 when the function is replaced with a relation. For the other direction, assume that $\mathcal{T}$ satisfies both **ATP** and **STP**. We first show that if R is approximately functionalisable then it is approximately determinisable when $\mathcal{T}$ (that defines R) satisfies both **ATP** and **STP**. Let f be a rational function that approximately functionalises R w.r.t. Levenshtein distance, i.e., $d(R, f) < \infty$. Thus, a functional transducer $\mathcal{F}$ that defines f satisfies $d(\mathcal{T}, \mathcal{F}) < \infty$. From Lemma 3.6, because $\mathcal{T}$ satisfies both **ATP** and **STP** and $d(\mathcal{T}, \mathcal{F}) < \infty$, it follows that $\mathcal{F}$ also satisfies **ATP** and **STP**. Using Lemma 4.8, the rational function f is approximately determinisable. Let g be a sequential function that approximately determinises f w.r.t Levenshtein, i.e, $d(f, g) < \infty$. Since d is a metric on relations, $d(R, g) \leq d(R, f) + d(f, g)$. Since both $d(R, f)$ and $d(f, g)$ are finite, we get $d(R, g) < \infty$. Hence, the rational relation R is approximately determinisable.

Now it suffices to show that R is approximately functionalisable w.r.t. Levenshtein family of distance. Towards this, we prove that $\mathsf{diff}_d(R) < \infty$ when $\mathcal{T}$ satisfies **ATP**. Observe that $\mathsf{diff}_d(R) = dia_d(R_o)$ where $R_o = \{(v_1, v_2) \mid \exists u \in dom(R), (u, v_1), (u, v_2) \in R\}$ is a rational relation that consists of all pairs of output words of R on any input. Using Lemma 4.5, $dia_d(R_o) \leq N_{\mathcal{T}}$ when $\mathcal{T}$ satisfies **ATP**. Hence, $\mathsf{diff}_d(R) = dia_d(R_o) < \infty$, and R is approximately functionalisable w.r.t. d by virtue of Lemma 5.3. ◀

Since **ATP** and **STP** are decidable for transducers (Lemma 3.7), we obtain the following.

▶ **Theorem 5.6.** *The approximate determinisation problem for rational relations given as rational transducers w.r.t. a metric $d \in \{d_l, d_{lcs}, d_{dl}\}$ is decidable.*

Approximate uniformisation problem. Given a relation $R \subseteq A^* \times B^*$, a *uniformiser* of R is a function $U : A^* \to B^*$ such that $dom(R) = dom(U)$ and for all $u \in dom(R)$, it holds that $(u, U(u)) \in R$. It is known that any rational relation admits a rational uniformiser [26], which is not true if we seek for a sequential uniformiser. Moreover, the problem of deciding whether a given rational relation admits a sequential uniformiser is undecidable [11]. We consider an approximate variant of this problem.

▶ **Definition 5.7** (Approximate Uniformisation)**.** *A rational relation R is* approximately uniformisable *w.r.t. a metric d if there exists a uniformiser U of R and a sequential function f such that $d(U, f) < \infty$. In that case, we say that R is* d-approximate uniformisable *by a sequential function.*

In other words, R is d-approximate uniformisable by a sequential function f iff there exists an integer $k \in \mathbb{N}$ such that for all $u \in dom(R)$, there exists $(u, v) \in R$ with $d(v, f(u)) \leq k$.

▶ **Theorem 5.8.** *Checking whether a rational relation is d-approximate uniformisable for $d \in \{d_l, d_{lcs}, d_{dl}\}$ is undecidable.*

Proof. Let $\phi_1 : \{1, \dots, k\} \to B^*$ and $\phi_2 : \{1, \dots, k\} \to B^*$ be two morphisms defining an instance of the post correspondence problem (PCP), which asks whether there exists $w \in \{1, \dots, k\}^+$ such that $\phi_1(w) = \phi_2(w)$. Let $A = B \uplus \{1, \dots, i, a, b, \#\}$.

Let R be the relation which as input takes any word of the form $w_1 \# w_2 \# \dots w_m \# X$ where $w_i \in \{1, \dots, k\}^*$ for $1 \leq i \leq m$ and $X \in \{a, b\}$. Let us define the outputs:

- If $X = a$, then the only output is $\phi_1(w_1)\#\phi_1(w_2)\# \dots \phi_1(w_m)\#$.
- If $X = b$, then any word of the form $v_1 \# \dots v_m \#$ where $v_i \neq \phi_2(w_i)$ for all $1 \leq i \leq m$ is a valid output.

It can be shown that R is rational, recognizable by some transducer $\mathcal{T}$. We now show that R is approx-uniformisable iff PCP has no solution iff R is exact-uniformisable. First, suppose that PCP has a solution w and R is approx-uniformisable by some sequential transducer $\mathcal{D}$, i.e. $d(\mathcal{T}, \mathcal{D}) \leq K$ for some K. Consider inputs of the form $u_\ell = (w\#)^\ell$ for $\ell \geq 0$, and let $\alpha_\ell, \alpha_a, \alpha_b$ be such that $q_0 \xrightarrow{u_\ell|\alpha_\ell}_{\mathcal{D}} q, q \xrightarrow{a|\alpha_a}_{\mathcal{D}} q_f$ and $q \xrightarrow{b|\alpha_b}_{\mathcal{D}} p_f$, where q_0 is the initial state, q is a state and q_f, p_f are final states of $\mathcal{D}$. Since $d(\mathcal{T}, \mathcal{D}) \leq K$, for all ℓ, $d((\Phi_1(w)\#)^\ell, \alpha_\ell\alpha_a) \leq K$ holds. Similarly, for all ℓ, there exist $v_1, v_2, \dots, v_\ell$ all different from $\Phi_2(w)$ such that

$$d(v_1\#v_2\# \dots v_\ell\#, \alpha_\ell\alpha_b) \leq K.$$

Since $d(\alpha_\ell\alpha_a, \alpha_\ell\alpha_b)$ is uniformly bounded for all ℓ by some M, by applying triangular inequalities, we get that for all ℓ, there exist $v_1, \dots, v_\ell$ all different from $\Phi_2(w)$ such that

$$d((\Phi_1(w)\#)^\ell, v_1\#v_2\# \dots v_\ell\#) \leq 2K + M$$

Take $\ell = 4K + 2M + 2$, and fix a sequence of at most $2K + M$ edits from $(\Phi_1(w)\#)^\ell = (\Phi_1(w)\#\Phi_1(w)\#)^{2K+M+1}$ to $v_1\#v_2\# \dots v_\ell\#$. In this sequence, at least one copy of $(\Phi_1(w)\#\Phi_1(w)\#)$ is not edited by the sequence. Therefore, there exists some i such that $\#v_i\# = \#\Phi_1(w)\#$ and hence $v_i = \Phi_1(w)$. It is a contradiction since $v_i \neq \Phi_2(w) = \Phi_1(w)$. Therefore R is not approximately uniformisable.

Conversely, if there is no solution to PCP, then the following sequential function is an exact uniformiser of R: on any input of the form $w_1\# \dots \#w_mX$ it outputs $\Phi_1(w_1)\# \dots \#\Phi_1(w_m)$. ◀

6 Future works

In this paper, we proved that approximate determinisation is decidable for functional transducers, by checking various twinning properties. We have shown that **HTP** and **STP** are decidable in PTime, which entails that approximate determinization of functional transducers for the Hamming distance is decidable in PTime. For the other distances, such as Levenshtein distance, the time complexity is doubly exponential, as deciding conjugacy of a rational relation, and hence **ATP**, is doubly exponential [1]. We conjecture that this is suboptimal, and leave as future work finding a better upper-bound.

We show that approximate uniformisation is undecidable for Levenshtein family. We leave the case of Hamming distance as future work. The current undecidability proof for Levenshtein family, based on PCP, heavily requires that the lengths of the output words produced on transitions can differ, which may not guarantee that the total output lengths are the same, which is necessary to have a finite Hamming distance. Our proof also does not extend to the letter-to-letter setting, where both the given transducer and the required uniformiser process *and produce* a single letter on every transition. This problem is closely

related to the standard Church synthesis for regular specifications, with the modification that the strategy to be synthesized is allowed to make a bounded number of errors. To our knowledge, this variant has not been studied in the literature.

Finally, we studied approximate decision problems up to finite distance. Another interesting question is to consider their "up to distance k" variant, where k is given as input.

References

1 C. Aiswarya, Amaldev Manuel, and Saina Sunny. Deciding conjugacy of a rational relation - (extended abstract). In *28th International Conference on Developments in Language Theory, DLT 2024*, volume 14791 of *Lecture Notes in Computer Science*, pages 37–50. Springer, 2024. doi:10.1007/978-3-031-66159-4_4.

2 C. Aiswarya, Amaldev Manuel, and Saina Sunny. Edit Distance of Finite State Transducers. In *51st International Colloquium on Automata, Languages, and Programming ICALP 2024*, volume 297 of *LIPIcs*, pages 125:1–125:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ICALP.2024.125.

3 Rajeev Alur, Adam Freilich, and Mukund Raghothaman. Regular combinators for string transformations. In *Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14*, pages 9:1–9:10. ACM, 2014. doi:10.1145/2603088.2603151.

4 Benjamin Aminof, Orna Kupferman, and Robby Lampert. Rigorous approximated determinization of weighted automata. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011*, pages 345–354. IEEE Computer Society, 2011. doi:10.1109/LICS.2011.50.

5 Benjamin Aminof, Orna Kupferman, and Robby Lampert. Rigorous approximated determinization of weighted automata. *Theoretical Computer Science*, 480:104–117, 2013. doi:10.1016/J.TCS.2013.02.005.

6 Udi Boker and Thomas A. Henzinger. Approximate determinization of quantitative automata. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012*, volume 18 of *LIPIcs*, pages 362–373. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPICS.FSTTCS.2012.362.

7 Udi Boker and Thomas A. Henzinger. Exact and approximate determinization of discounted-sum automata. *Logical Methods in Computer Science*, 10(1), 2014. doi:10.2168/LMCS-10(1:10)2014.

8 Adam L. Buchsbaum, Raffaele Giancarlo, and Jeffery R. Westbrook. An approximate determinization algorithm for weighted finite-state automata. *Algorithmica*, 30(4):503–526, 2001. doi:10.1007/S00453-001-0026-6.

9 Marie-Pierre Béal and Olivier Carton. Determinization of transducers over finite and infinite words. *Theoretical Computer Science*, 289(1):225–251, 2002. doi:10.1016/S0304-3975(01)00271-7.

10 Marie-Pierre Béal, Olivier Carton, Christophe Prieur, and Jacques Sakarovitch. Squaring transducers: an efficient procedure for deciding functionality and sequentiality. *Theoretical Computer Science*, 292(1):45–63, 2003. doi:10.1016/S0304-3975(01)00214-6.

11 Arnaud Carayol and Christof Loeding. Uniformization in automata theory. In *14th International Congress of Logic, Methodology and Philosophy of Science*, pages 153–178. London: College Publications, 2015.

12 Christian Choffrut. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoretical Computer Science*, 5(3):325–337, 1977. doi:10.1016/0304-3975(77)90049-4.

13 Christian Choffrut and Serge Grigorieff. Uniformization of rational relations. In *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, pages 59–71. Springer, 1999.

14 Christian Choffrut and Marcel Paul Schützenberger. Décomposition de fonctions rationnelles. In *3rd Annual Symposium on Theoretical Aspects of Computer Science STACS 1986*, volume 210 of *Lecture Notes in Computer Science*, pages 213–226. Springer, 1986. doi:10.1007/3-540-16078-7_78.

15 Samuel Eilenberg. *Automata, languages, and machines. vol. A*. Pure and applied mathematics. Academic Press, 1974.

16 Diego Figueira and Leonid Libkin. Path logics for querying graphs: Combining expressiveness and efficiency. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015*, pages 329–340. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.39.

17 Emmanuel Filiot, Ismaël Jecker, Christof Löding, and Sarah Winter. On equivalence and uniformisation problems for finite transducers. In *43rd International Colloquium on Automata, Languages, and Programming ICALP 2016*, volume 55 of *LIPIcs*, pages 125:1–125:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.ICALP.2016.125.

18 Emmanuel Filiot, Nicolas Mazzocchi, and Jean-François Raskin. A pattern logic for automata with outputs. *International Journal of Foundations of Computer Science*, 31(6):711–748, 2020. doi:10.1142/S0129054120410038.

19 Emmanuel Filiot and Pierre-Alain Reynier. Transducers, logic and algebra for functions of finite words. *ACM SIGLOG News*, 3(3):4–19, 2016. doi:10.1145/2984450.2984453.

20 Seymour Ginsburg and Gene F. Rose. A characterization of machine mappings. *Journal of Symbolic Logic*, 33(3):468–468, 1968. doi:10.2307/2270340.

21 Eitan M. Gurari and Oscar H. Ibarra. A note on finitely-valued and finitely ambiguous transducers. *Mathemtical Systems Theory*, 16(1):61–66, 1983. doi:10.1007/BF01744569.

22 Thomas A. Henzinger, Jan Otop, and Roopsha Samanta. Lipschitz robustness of finite-state transducers. In *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014*, volume 29 of *LIPIcs*, pages 431–443. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2014. doi:10.4230/LIPICS.FSTTCS.2014.431.

23 J. Berstel. *Transductions and Context-Free Languages*. Teubner, Stuttgart, 1979.

24 Ismaël Jecker and Emmanuel Filiot. Multi-sequential word relations. *International Journal of Foundations of Computer Science*, 29(2):271–296, 2018. doi:10.1142/S0129054118400075.

25 Daniel Kirsten and Sylvain Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In *26th International Symposium on Theoretical Aspects of Computer Science STACS 2009*, volume 3 of *LIPIcs*, pages 589–600. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany, 2009. doi:10.4230/LIPICS.STACS.2009.1850.

26 K. Kobayashi. Classification of formal languages by functional binary transductions. *Information and Control*, 15(1):95–109, 1969. doi:10.1016/S0019-9958(69)90651-2.

27 Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997. URL: https://aclanthology.org/J97-2003.pdf.

28 Anca Muscholl and Gabriele Puppis. The many facets of string transducers (invited talk). In *36th International Symposium on Theoretical Aspects of Computer Science STACS 2019*, volume 126 of *LIPIcs*, pages 2:1–2:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICS.STACS.2019.2.

29 George N. Raney. Sequential functions. *Journal of the ACM*, 5(2):177–180, 1958. doi:10.1145/320924.320930.