The Many Facets of String Transducers

Anca Muscholl
LaBRI, University of Bordeaux, France

Gabriele Puppis
CNRS, LaBRI, France

Abstract

Regular word transductions extend the robust notion of regular languages from a qualitative to a quantitative reasoning. They were already considered in early papers of formal language theory, but turned out to be much more challenging. The last decade brought considerable research around various transducer models, aiming to achieve similar robustness as for automata and languages. In this paper we survey some older and more recent results on string transducers. We present classical connections between automata, logic and algebra extended to transducers, some genuine definability questions, and review approaches to the equivalence problem.

2012 ACM Subject Classification Theory of computation → Transducers

Keywords and phrases String transducers, complexity

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.2

Category Invited Talk

Funding Work supported by ANR DeLTA (ANR-16-CE40-0007).

Acknowledgements We would like to thank our colleagues, in particular D. Figueira, O. Gauwin, F. Mazowiecki and I. Walukiewicz, for their comments on a draft version of this paper.

1 Introduction

Since the early times of computer science, the notion of transduction has played a fundamental role, as computers typically process data and transform it between different formats. Numerous fields of computer science are ultimately concerned with transformations, ranging from databases to image processing, and an important issue is to perform transformations with low cost, whenever possible.

The most basic form of transformations is realized by processing inputs and producing outputs using finite memory. Such machines are called finite-state transducers. Finite-state string transducers were considered in very early work in formal language theory [22, 73, 41, 54, 67, 1, 40, 26, 14], and it was soon clear that achieving a good understanding of transducers would be much more challenging than for the classical finite-state automata. There are many aspects that change from automata to transducers, in particular non-determinism and the capability to process the input in both directions strictly increase the expressive power of transducers, while this not the case for automata [69, 77]. A further difference is that some fundamental questions, such as the equivalence problem, are undecidable for transducers.

We consider in this survey string transducers, as non-deterministic finite-state transducers that compute relations (or functions) over finite words. It turns out that for string-to-string functions, (single-valued) two-way transducers nicely capture the notion of regularity: Engelfriet and Hoogeboom showed in [42] that two-way transducers have the same expressive power as Courcelle's monadic second-order logic definable graph transductions [28], restricted to words. This equivalence supports thus the notion of "regular" functions, in the spirit of classical results on regular word languages from automata theory and logic (due to

Büchi, Elgot, Trakhtenbrot, Rabin, and others). Recently, Alur and Cerný [3] fostered new interest into this topic by introducing streaming string transducers, that have the same expressiveness as the two previous models, but allow for more flexibility in extending the model of string-to-string transductions to other quantitative models.

This survey is about classical connections between automata, logic and algebra extended to transducers, some genuine definability questions for transducers, and the main approaches to the equivalence problem. For space reasons we had to leave out several interesting topics, for example regular expressions for transductions [6, 33, 20], transductions over infinite strings [5, 45], visibly pushdown transducers [31, 49], or multi-tape transducer models [63, 25, 23]. Finally, we acknowledge the inspiration provided by a recent survey by Filiot and Reynier [50].

2 String transducers: the setting

As it happens for languages, relations on strings can be specified using many different formalisms. Following a long-lasting tradition, classes of relations are defined using equational, descriptive, or operational formalisms, e.g. algebras, logics, or automata. Unlike for languages, however, different specification formalisms for relations have often different expressive powers, as well as different closure and algorithmic properties. Another important distinction is whether we consider arbitrary relations or functions. For simplicity, hereafter we use the term *function* to generically refer to a partial function, and we say that a relation is *single-valued* if it is a (partial) function. Similarly, we say that a relation is *k-valued* (resp. *finite-valued*) is it is a union of k (resp. finitely many) partial functions.

Below, we present a few common formalisms based on logics and automata that received most of the attention in the literature.

MSO transductions

Within the realm of logics, monadic second-order (MSO) logic is one of the most expressive formalisms for specifying functions, but also relations on words. The approach, as originally proposed by Courcelle for graphs [28], consists of logically interpreting an output on a given input (or copies of it), where both input and output are seen as relational structures. More precisely, an MSO interpretation consists of a family of MSO formulas φ_{dom}, $\varphi_{\text{pos}}(x)$, $\varphi_{c\text{-lab}}(x)$ (one for every letter c in the target alphabet), and $\varphi_{\text{succ}}(x, y)$, that describe, respectively, when the output string is defined, which positions from the input become positions of the output, the labels of the output positions, and the successor relation. An *MSO transduction* can be seen as an MSO interpretation preceded by an operation that copies the input a fixed number of times, annotating each copy with a distinguished color. For example, Figure 1 gives a very simple example of MSO transduction that removes all occurrence of b from the input, and replaces every a by a c.

$$\varphi_{\text{pos}}(x) \;:=\; a(x)$$

$$\varphi_{c\text{-lab}}(x) \;:=\; \texttt{true}$$

$$\varphi_{\text{succ}}(x,y) \;:=\; x < y \;\wedge\; a(x) \;\wedge\; a(x) \;\wedge\; \forall z \;\; x < z < y \rightarrow b(z)$$

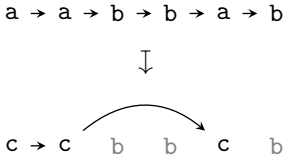


Figure 1 An MSO transduction with formulas defining positions, labels, and successor relation.

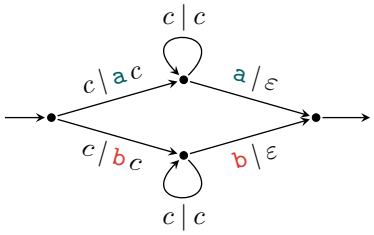


Figure 2 A 1NFT. An arrow labeled by $c \mid v$ represents a transition that consumes an input letter $c \in \{a, b\}$ and appends the word v to the output.

In order to define multi-valued relations, one can first annotate the input with arbitrary colors, and then perform an MSO transduction on the annotated input: this allows a certain degree of freedom in the possible outputs associated with the original input. The latter type of relation is called *NMSO transduction*, the 'N' standing for non-deterministic. There are of course restricted variants of (N)MSO transductions, where, for instance, monadic second-order quantification is forbidden (*FO transductions*), or where the order on output positions is directly inherited from the order on input positions (*order-preserving MSO transductions*). For example, the transduction of Figure 1 is an order-preserving FO transduction.

While the formalism of string-to-string (N)MSO transductions is already quite expressive, automata are certainly the most versatile tool for defining relations in an operational way. The literature offers many different models of automata for relations (a.k.a. transducers), which can roughly be distinguished based on the amount of non-determinism and on how the input and the associated output are consumed and produced, respectively. For instance, one can distinguish between models in which the input head moves only from left to right (one-way) or in either direction (two-way).

One-way transducers

A rather simple model is that of *one-way non-deterministic finite-state transducer* (abbreviated as 1NFT), describing the so-called *rational relations*. This is basically a non-deterministic automaton in which every transition consumes at most one letter from the input and appends a word of any length to the output. Figure 2 gives an example of a 1NFT realizing the cyclic rotation f defined by $f(ua) = au$ and $f(ub) = bu$, for all $u \in \{a, b\}^*$.

1NFT are readily seen to be equivalent to the formalism of order-preserving NMSO transductions. They can also be equally presented as languages. For example, Nivat [67] observed that the set of runs of a 1NFT can be presented as a language of interleavings of input and output letters, called synchronization language. As a consequence, every rational relation $R \subseteq \Sigma^* \times \Gamma^*$ can be presented by at least one regular synchronization language (possibly more) over $\Sigma \uplus \Gamma$.

The deterministic variant of 1NFT, denoted 1DFT and defining the so-called of *sequential functions*, has a deterministic underlying automaton and at most one transition (p, a, q, u), for every pair of states p, q and input letter a. In addition, every final state is associated with an output word that is produced at the end of the run[1].

[1] This feature is necessary to realize functions that are not monotone w.r.t. prefix order. According to the classical terminology, these functions should be called subsequential, whereas sequential was originally

Figure 3 A DSST that updates two registers, named x, y, based on whether the input letter c is the blank space or not.

As a matter of fact, interesting subclasses of rational relations can be obtained by restricting the forms of interleavings of input and output letters in the synchronization languages. For example, synchronization languages contained in $(\Sigma\Gamma)^*(\Sigma^* \cup \Gamma^*)$, which enforce a strict alternation between input and output letters, capture the class of automatic relations. This latter class is closed under all boolean operations (union, intersection, complement), and has a decidable equivalence problem and several other nice algorithmic properties [71]. The idea of defining classes of relations based on synchronization languages was investigated in detail in a series of works [43, 47, 36, 35].

Two-way transducers

The *two-way* variant of a non-deterministic finite-state transducer (abbreviated as 2NFT) allows the input head to move in any direction, to the left or to the right. This gives a more powerful model than 1NFT, which captures e.g. the relation $\{(u, u^n) \ : \ u \in \Sigma^*, n \in \mathbb{N}\}$.

When 2NFT are restricted to be single-valued, they turn out to be be equivalent to MSO transductions, so to their deterministic variant 2DFT as well [42]. This is a striking difference compared to the one-way models, where single-valued 1NFT are strictly more powerful than 1DFT.

Streaming string transducers

More recently, a third transducer model, called *streaming string transducer* (abbreviated as DSST or NSST, depending on whether it is deterministic or not), was proposed by Alur and Cerný in [3]. In this model the input is processed from left to right, while storing partial outputs in a finite set R of write-only registers. Transitions consume one input letter at a time, and can append words to the left and to the right of a register, as well as concatenate registers together. In particular, register updates correspond to substitutions $\sigma : R \to (R \cup \Gamma)^*$, where Γ is the output alphabet. As an example, the streaming transducer in Figure 3 reformats author names in bibliographies by transforming strings of the form `first-name␣second-name␣surname` into strings of the form `surname,␣first-name␣second-name`.

An SST is called *copyless* if no register occurs more than once in the right hand–=-sides of the update rules. According to the usual nomenclature we refer to copyless SST just as SST, and to unrestricted SST as *copyful SST*. Copyful DSST are close to an old formalism for grammars called HDT0L, a special family of Lindenmayer systems. Strictly speaking, HDT0L systems define word languages, as images of an initial word via sequences of morphisms, that is, words of the form $h_{i_1} \circ \cdots \circ h_{i_n}(w)$ for some indexes $i_1, \ldots, i_n$ and for a fixed tuple of morphisms $h_1, \ldots, h_k$ and a fixed word w. Such grammars can be extended naturally so as to define functions from an indexing sequence $i_1, \ldots, i_n$ (the input) to $h_{i_1} \circ \cdots \circ h_{i_n}(w)$

reserved to prefix-monotone functions realized by 1DFT without the final output rule. We prefer the generic name “sequential function”.

(the output). Copyful DSST define precisely those functions that can be obtained from the previous ones by restricting their domains to regular languages, and by possibly applying a final additional morphism [51].

In the single-valued case, NSST can be determinized and are expressively equivalent to 2DFT and MSO transductions [3]. Determinization incurs an exponential blow-up, and so does the transformation from DSST to 2DFT. Surprisingly, the reverse transformation from DSST to 2DFT turned out to be doable in quadratic time [32], through a construction based on reversible (i.e. deterministic and co-deterministic) transducers. Based on the equivalence between 2DFT, DSST, and MSO transductions, one often calls the induced class of string-to-string functions *regular*, in the spirit of classical results on regular languages, relating automata theory and logics. Unfortunately, this correspondence cannot be fully generalized to the relational case, where the transducer models 2NFT and NSST turn out to be incomparable. For example, the relation $\{(u, u^n) \ : \ u \in \Sigma^*, n \in \mathbb{N}\}$ is 2NFT-definable, but not NSST-definable, whereas the relation $\{(uv, vu) \ : \ u, v \in \Sigma^*\}$ is NSST-definable but not 2NFT-definable. It is however the case that NSST and NMSO transductions are equally expressive even in the relational case, as shown in [4]. Moreover, it is possible to capture the latter class of relations by a variant of two-way transducers, enhanced with the so-called *common guess* [19]. Formally, a two-way transducer (input-deterministic or not) has common guess if, before starting the computation, it can annotate the input with arbitrary colors, so that the same color is read each time the head revisits a position. This model is naturally closed under projections on the input, and easily simulates NSST. As a consequence, 2DFT with common guess are equivalent to NSST and to NMSO transductions:

▶ **Proposition 1.** *NMSO transductions, NSST, and 2DFT with common guess define the same class of relations.*

In the following sections we will consider two families of decision problems on relations: the class-membership and the equivalence problems. These are very natural problems, that cover most of the difficulties of reasoning with relations. As we will see, the rule of thumb in this context is that most problems turn out to be undecidable as soon as they involve non-trivial properties of *rational relations*. However, decidability can be recovered in restricted settings, notably in the single-valued, and possibly finite-valued, case. An alternative idea that is often used for recovering decidability of those problems is the origin semantics [17], that we discuss in Section 5.

3 Class-membership problems

Given any two classes $\mathcal{C}_1, \mathcal{C}_2$ of relations, the following class-membership (or characterization) problem arises naturally: *given a relation $R \in \mathcal{C}_1$, decide whether $R \in \mathcal{C}_2$*. Clearly, the decidability status and complexity of the above problem depends on the choice of the classes $\mathcal{C}_1, \mathcal{C}_2$, and on the formalisms used to represent their relations. Before discussing in more detail a few decidable cases, and their complexity, we give a necessary, and rather strict criterion for avoiding undecidability. The criterion is based on the following undecidability result for the universality problem of rational relations:

▶ **Theorem 2** (Fischer and Rosenberg [52]). *It is undecidable whether a given 1NFT realizes the universal relation $\Sigma^* \times \Gamma^*$.*

Proof. We give a simple proof of this undecidability result due to Ibarra [60], showing that undecidability even holds for a unary output alphabet. The proof reduces the Post

Correspondence Problem: given two word morphisms $f, g : \Omega^* \to \Delta^*$, with Ω and Δ finite alphabets, decide whether there is a non-empty word $w \in \Omega^+$ so that $f(w) = g(w)$. Given such f and g, let R_+ be the set of all pairs $(w\,u, c^n)$ such that $n = |u|$ and $u = f(w) = g(w)$ – here we assume w.l.o.g. that the alphabets Ω and Δ are disjoint. Intuitively R_+ consists of correct encodings of solutions of the PCP instance. Further let $R_- = (\Sigma^* \times \Gamma^*) \setminus R_+$ be the complement relation, where $\Sigma = \Omega \uplus \Delta$ and $\Gamma = \{c\}$. In particular, R_- contains pairs of the form $(w\,u, c^n) \in (\Omega^+\Delta^*) \times \Gamma^*$ such that either $n \neq |u|$, or $u \neq f(w)$, or $u \neq g(w)$. Note that R_- is universal if and only if there is no solution to the PCP instance.

Now, it suffices to verify that R_- is a rational relation. The pairs outside $(\Omega^+\Delta^*) \times \Gamma^*$ can be easily realized by a 1NFT. Similarly, the pairs $(w\,u, c^n)$ satisfying $n \neq |u|$ can be obtained by consuming the suffix u of the input while producing an output c^n of length strictly smaller or greater than $|u|$. The remaining pairs satisfying $u \neq f(w) \wedge n = |u|$ (resp. $u \neq g(w) \wedge n = |u|$) are covered using the following strategy. One guesses factorizations of w and u of the form $w_1\,a\,w_2$ and $u_1\,u_2$, so that u_2 does not begin with $f(a)$ (resp. $g(a)$), and $|u_1| = |f(w_1)|$ (resp. $|u_1| = |g(w_1)|$). Accordingly, one outputs $c^{|f(w_1)|}$ (resp. $c^{|g(w_1)|}$) while reading w_1, and $c^{|u_2|}$ while reading u_2. ◀

When considering a class-membership problem from a class $\mathcal{C}_1$ that contains all rational relations to a proper subclass $\mathcal{C}_2$ of it that contains at least the universal relation, one can sometimes adapt the proof above to show that the considered problem is undecidable. For example, following a result from [7], if $\mathcal{C}_1$ is the class of 2NFT-definable relations and $\mathcal{C}_2$ is the class of rational relations, then, given a PCP instance (f, g), one can modify the relation R_- in the previous proof by replacing the pairs $(w\,u, c^n)$ with pairs of the form $(w\,u, w\,c^n)$, where the first part w of the input is copied once to the output. These pairs can be easily produced by a two-way transducer that reads the input twice. It can be shown that, when the PCP instance has a solution, the second pass on the input is necessary for producing the correct number of c's. Otherwise, if the PCP instance has no solutions, then the relation becomes 1NFT-definable: a transducer can read the input prefix w to copy it onto the output, and then cover the arbitrary remaining parts of the input and the output. From this it follows that the class-membership problem from 2NFT to 1NFT is undecidable.

Theorem 2 exploits in a crucial way relations that associate arbitrarily many outputs with the same input, and in particular the existence of the universal relation. This suggests that decidability of universality and class-membership problems can be recovered by restricting to classes of functions. [2] We present below a few cases in which this approach succeeds, notably, for 1NFT/2NFT/NSST vs. single-valued 1NFT/2NFT/NSST (single-valuedness), for 1NFT vs. 1DFT (sequentiality), for 2NFT vs. single-valued 1NFT (one-way definability), and for 2NFT vs. FO transductions (FO-definability).

Single-valuedness

The class-membership problem from the class $\mathcal{C}_1$ of rational relations to the class $\mathcal{C}_2$ of rational functions boils down to testing single-valuedness of 1NFT. This property was first shown to be decidable by Schützenberger [75]. Later, polynomial-time algorithms were given in [58, 81, 12]. The precise complexity is NLOGSPACE-complete, which can be shown using a rather simple reduction to emptiness of one-counter automata, as explained below. Using

[2] Hereafter, for simplicity, we forbid the use of ε-moves in all models of one-way and two-way transducers (SST already forbid these moves by definition). This assumption does not affect the expressiveness of the models, nor the complexity of the considered problems, since in the single-valued case one can efficiently remove ε-moves.

a similar technique, one can prove that single-valuedness remains decidable for 2NFT and NSST, but now with PSPACE complexity (for 2NFT decidability was first shown in [30], and for NSST was shown in [4]).

▶ **Theorem 3.** *The single-valuedness problem is*
1. NLOGSPACE*-complete for 1NFT.*
2. PSPACE*-complete for 2NFT.*
3. *in* PSPACE *for NSST.*

Proof. Given a 1NFT T, one guesses an input word, together with two runs of T on it, and checks that the respective outputs are different. The test for different outputs can be done with a counter, that verifies on a given input that either (1) the lengths of the two outputs are different, or (2) there is some position in the two outputs where the respective symbols differ. In both cases the question can be reduced to emptiness of a one-counter automaton of quadratic size, which yields the claimed complexity. More precisely, the counter is updated on the basis of the number of output symbols produced by the transitions in the two runs, one contributing positively and the other contributing negatively; checking the same position in the two outputs amounts at checking that the value of counter is zero.

The result for single-valuedness of a 2NFT T follows the same idea as above, except that now the (one-way) one-counter automaton is obtained through a crossing sequence construction [77], and thus has exponential size. More precisely, one follows two runs of T on the same input by guessing tuples of states crossing each position. The length of every tuple can be bounded by $2n$, where n is the number of states of T, since to detect a violation of single-valuedness it suffices to consider only runs that visit the same position with the same state at most twice. This reduces single-valuedness of T to emptiness of a one-counter automaton of exponential size, which can be checked in PSPACE. Then PSPACE-hardness follows by reducing emptiness of intersection of finite-state automata [62].

Similarly, for NSST, the one-counter automaton that checks different outputs is exponential in the number of registers, since it needs to guess, for instance, which registers have a non-empty content that eventually appears in the final output [4]. To the best of our knowledge no matching lower bound is known in this case. ◀

Sequentiality

Let us now consider the problem of testing sequentiality of 1NFT, namely, the class-membership problem from rational to sequential functions. The problem was first shown decidable by Choffrut [26]. Later the complexity was shown to be in PTIME [11]. In fact, a close inspection to those proofs shows that the problem is NLOGSPACE-complete:

▶ **Theorem 4.** *The sequentiality problem for 1NFT is* NLOGSPACE*-complete.*

Proof. The theorem is established by first giving a topological characterization of sequential functions as those rational functions that are *Lipschitz w.r.t. the prefix distance* d; more precisely, such that there is a uniform constant k satisfying $d(f(u), f(v)) \leq k \cdot d(u, v)$ for all $u, v \in dom(f)$, where $d(u, v) = |u| + |v| - 2|u \wedge v|$ and $u \wedge v$ is the longest common prefix of u and v. For example, the function $f : cu \mapsto uc$ ($c \in \{a, b\}$, $u \in \{a, b\}^*$) is Lipschitz with constant $k = 1$, while its inverse $f^{-1} : uc \mapsto cu$ is not.

If one starts with a 1NFT T, the above characterization can be rephrased in terms of a structural property of the squared transducer T^2, which has for states the pairs of states of T and associates with any input u a pair of possible outputs (v_1, v_2) of T on u. Formally

[11], one proves that the function realized by T is Lipschitz w.r.t. the prefix distance if and only if T^2 satisfies the so-called *twinning property*:

- for every path $(q_0, q_0') \xrightarrow{u|v_1,v_2} (q, q') \xrightarrow{v|w_1,w_2} (q, q')$ in T^2, with (q_0, q_0') initial state, either $w_1 = w_2 = \varepsilon$ or $|w_1| = |w_2| \ \wedge \ v_1 \cdot w_1^\omega = v_2 \cdot w_2^\omega$ (in particular, the latter condition implies that w_1, w_2 are conjugated).

For non-empty w_1, w_2 as above, one can check the condition $|w_1| = |w_2|$ of the twinning property with a one-counter automaton of quadratic size. As for the second condition $v_1 \cdot w_1^\omega = v_2 \cdot w_2^\omega$, it boils down to checking that the trimmed part of T, seen as a transducer on infinite words with a trivial Büchi condition (all states set to be final), is single-valued. The latter problem is easily shown to be in NLOGSPACE. ◀

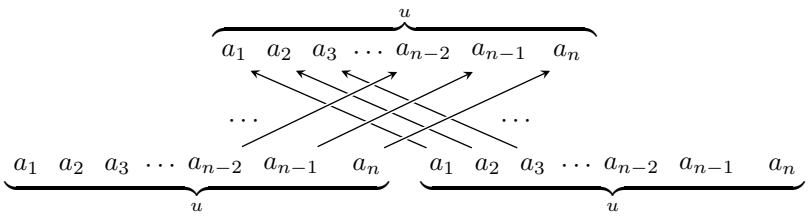
Some generalizations of the above result were given for transducers on infinite ω-words [11] and for variants of SST with updates of the form $x := y \cdot v$, where x, y are registers and v is a word (possibly over an infinitary group) [34].

One-way definability

Here we explain how to decide which regular functions are rational, or, equally, we solve the class-membership problem from single-valued 2NFT to 1NFT. The problem was first shown decidable, with non-elementary complexity, by Filiot et al. in [46]. In [9, 10], the complexity was improved to 2-EXPSPACE, together with an EXPSPACE lower bound. A refinement of a pumping argument in [10] due to I. Jecker, shows that the problem is EXPSPACE-complete.

▶ **Theorem 5.** *One-way definability of 2NFT is* EXPSPACE*-complete.*

The key notion underlying the above result is that of *inversion* of a run, which roughly corresponds to having long factors of the output that are generated without following the left-to-right order. The minimum length of factors forming an inversion is chosen as a function of the number of states of the transducer, so as to enable suitable pumping and combinatorial arguments. The characterization shows that a single-valued 2NFT is equivalent to some 1NFT if and only if every inversion in every successful run delimits a factor of the output that is periodic, with uniformly bounded period. As an example, consider a 2NFT T that realizes the function $f(u) = uu$, e.g. as follows (we draw arrows to represent some dependency relationships between positions in the output and in the input):



Every run of T that is long enough must have an inversion, essentially because one can find many output positions in the first copy of u and many output positions in the second copy of u for which the origin arrows are pairwise crossing. Based on the previous characterization, we know that f cannot be defined by a 1NFT. Observe however, that if the domain of f is restricted to a periodic language, e.g. $dom(f) = (ab)^*$, then f becomes definable by a 1NFT, e.g. one that produces ab at each position of the input.

In [8] a result similar to Theorem 5 is obtained, that characterizes effectively the functions definable by sweeping transducers, i.e., 2NFT with head reversals occurring only at the extremities of the input. This latter characterization can be used to minimize the number of

passes performed by a sweeping transducer. Moreover, [8] shows a correspondence between the number of passes of sweeping transducers and the number of registers of equivalent *concatenation-free* NSST, namely, NSST in which the right hand-side of every update rule contains at most one register (in particular, updates of the form $x := \dots y \dots z \dots$ are forbidden). Based on this correspondence one derives a minimization procedure for the number of registers of concatenation-free NSST:

▶ **Theorem 6** ([8]). *One can compute in 2-*Expspace *the minimum number of registers needed to implement a function given as a concatenation-free NSST. The complexity is* Expspace *if the given NSST is unambiguous.*

FO-definability

We finally turn towards FO-definability of regular functions. The reader may recall the deep theorem of Schützenberger about the equivalence of star-free and aperiodic word languages [74]. This theorem gives a decision procedure for knowing whether a regular language is first-order expressible (or equivalently, [64]), using semigroup properties (see also [68, 82, 38, 39] for some more recent presentations). The general idea is to lift the characterization of FO-definable regular languages to regular functions.

A first trivial observation is that FO-definability of the domain language is a necessary, but not a sufficient condition: for instance, the function $f(w) = w(2)w(4)\cdots w(2\lfloor \frac{|w|}{2} \rfloor)$ has domain Σ^*, which is a star-free language, but it is not an FO transduction. We will see that the natural and correct choice is rather to test aperiodicity on the automaton underlying the transducer – intuitively, aperiodicity means that any two inputs u^n and u^{n+1}, for large enough n, induce the same transformation in the underlying automaton.

The FO-definability problem for regular functions is still open in its full generality, and thought to be difficult. Non-effective characterizations were obtained for 2NFT [24] and for DSST [48]. In both cases the characterization is expressed in terms of the aperiodicity property of a suitable transition monoid – for 2NFT, this is just the standard transition monoid of the underlying automaton, while for DSST, the transition monoid is defined on the basis of the underlying automaton and the register updates.

▶ **Theorem 7** ([24, 48]). *The string-to-string FO transductions are precisely*
- *the functions realized by 2NFT with an aperiodic transition monoid,*
- *the functions realized by DSST with an aperiodic transition monoid.*

▶ **Open question 8.** *Is it decidable whether a transduction given by a 2NFT/DSST is first-order definable?*

There are essentially two ways by means of which one can simplify the FO-definability problem and establish decidability: by either moving from classical semantics to origin semantics, or by restricting further the class of functions. The first approach was followed in [17], where the problem of deciding whether a given 2DFT is origin-equivalent to some FO transduction was shown to be decidable. We will discuss this approach later, in Section 5, while here we focus on the second approach, specifically by considering the subclass of rational (rather than regular) functions. We point out that both approaches, in the end, give a notion of 'minimal' object that is used as a canonical representative of the given function.

For the sake of presentation, we begin by considering an even more restricted case, that of sequential (i.e. 1DFT-definable) functions. As already explained, the goal here is to test aperiodicity on the automaton underlying the minimal 1DFT, where minimality is intended

in a strong categorical sense, namely, as an object to which every other equivalent object can be homomorphically mapped to:

▶ **Theorem 9** ([27]). *For every sequential function f, there is a transducer T_f that realizes f and such that for every equivalent trimmed transducer T, there is a unique morphism from T to T_f. Moreover, T_f can be constructed in* PTIME *from a 1DFT realizing f.*

Before sketching the construction of the minimal transducer T_f, it is worth spending a few more words on the notion of transducer morphism. For this notion, it is tempting to adopt the classical definition of graph morphism, that preserves transitions as labeled edges. However, there are 1NFT that realize the same function using a minimal number of states, but that are not isomorphic in the usual sense, e.g.:

This simple observation suggests that the correct notion of transducer morphism should take into account the possible different 'speeds' at which equivalent transducers may produce their outputs. This can be done by pairing the morphism application with a mapping from states to elements of the free group $(\Gamma \cup \Gamma^{-1})^*$ that encode output lags.

Proof of Theorem 9. The main ingredient underlying the definition of T_f is a notion of congruence for sequential functions, very similar to the Nerode congruence. The additional crux is to correctly identify words that induce the same behavior w.r.t. the produced output. This requires a normalization step, that guarantees that outputs are produced *as early as possible*, based only on the finite information that comes from the consumed part of the input. Formally, given a function f, let $\hat{f}$ be the function that has as domain the prefix closure of the domain of f, and such that $\hat{f}(u) = \bigwedge\{f(uv) \ : \ uv \in dom(f)\}$ is the longest common prefix of all $f(uv)$. Using $\hat{f}$, one defines the equivalence $\equiv_f$ by $u \equiv_f v$ if and only if for all words w, either $f(uw)$ and $f(vw)$ are undefined, or they are both defined and, once the respective prefixes $\hat{f}(u)$ and $\hat{f}(v)$ are removed, they result in the same word. It is not difficult to see that $\equiv_f$ is a right congruence, and that it has finite index if and only if f is sequential. Finally, one defines[3] the minimal transducer T_f that has as states the $\equiv_f$-equivalence classes $[u]$ ($u \in \Sigma^*$) and transitions of the form $[u] \xrightarrow{a|v} [ua]$, where v is obtained from $\hat{f}(ua)$ by removing the prefix $\hat{f}(u)$. ◀

Based on Theorem 7 and Theorem 9, one concludes that f is an FO transduction if and only if the underlying automaton of T_f is finite and aperiodic. Moreover, it is easy to see that if f is defined by a 1DFT, then T_f is finite, and aperiodicity can be tested in PSPACE. This shows that the FO-definability problem for 1DFT is in PSPACE.

Recently, Filiot et al. [44] generalized the characterization of FO-definability from 1DFT to 1NFT (in fact, to the equivalent, but less succinct model of bimachines):

▶ **Theorem 10** ([44]). *A rational function f is an FO transduction if and only if its canonical bimachine is aperiodic. Moreover, if f is given by a bimachine (resp. by a 1NFT), then the property can be decided in* PSPACE *(resp. 2-*EXPTIME*).*

[3] Technically speaking, the outlined definition of T_f misses the production of the initial prefix $\hat{f}(\varepsilon)$ of the output. To fix this problem one can equip transducers with an initial production rule, very similar to the final production rule that they already have (cf. [27]).

We discuss the terminology and the main ideas underlying the decidability of the above problem (for more details, we refer the reader to [44, 50]). First, the model of bimachine, which was originally introduced in [74], is conveniently seen as a 1DFT enhanced with *regular look-ahead*. Here by regular look-ahead we mean a finite-state automaton that deterministically processes the input from right to left (so it is co-deterministic). The transitions of a 1DFT T enhanced with the look-ahead automaton A depend on the current state of T, the input symbol, and the state reached by A after processing the suffix of the input up to the current position. In [41], it is shown that 1NFT are exactly as expressive as bimachines (or equally, 1DFT with regular look-ahead), with a doubly-exponential time translation from the former to the latter – one exponent is due to the co-deterministic look-ahead, and the other exponent is due to the determinization of the transducer parametrized by the look-ahead.

The existence of a canonical bimachine realizing a rational function f essentially relies on the possibility of computing a minimal look-ahead automaton based only on f, and then using this to reduce to the previous minimization problem for simple 1DFT without look-ahead. More precisely, in [70] it is shown that, given any function f, one can define a co-deterministic (possibly infinite) automaton A_f such that f is rational if and only if

- A_f is finite,
- the function $f[A_f]$ that maps every input $u \in dom(f)$, annotated with the unique run of A on u, to the output $f(u)$, is sequential.

For example, the co-deterministic automaton A_f can be obtained from a left congruence $\sim_f$ defined by $u \sim_f v$ if and only if there is a bound k (depending on u, v) such that, for all words w, either $f(wu), f(wv)$ are undefined, or they are both defined and $d(f(wu), f(wv)) \leq k$, where d is the prefix distance defined on page 7.

Summing up, assuming that f is rational, one constructs the minimal co-deterministic automaton A_f and the minimal 1DFT T_f for the sequential function $f[A_f]$. Pairing A_f and T_f results in a minimal and canonical bimachine realizing f. It is then routine to prove that f is an FO transduction if and only if both A_f and T_f are aperiodic.

We conclude by observing that the complexity in Theorem 10 is optimal when we are concerned with functions described by bimachines (PSPACE-hardness follows from the usual reduction from universality of finite-state automata). However, when the function is given by a 1NFT, the FO-definability problem is only known to be between PSPACE and 2-EXPTIME.

4 The equivalence problem

Historically, string transducers have been first studied within their deterministic, or unambiguous, one-way machine models. One reason for this is that many fundamental problems about rational relations are undecidable, in particular the equivalence problem (recall Theorem 2).

This section reviews the status of the equivalence problem for various types of string transducers, together with a selection of the main technical arguments for proving decidability. One of the first references for decidability in the deterministic case, and specifically for length-preserving 1DFT, is due to Moore [66]. The result was then extended progressively. Equivalence for 1DFT was shown decidable by Blattner and Head [16]. The same authors also showed that it is decidable whether a 1NFT is single-valued (see also [75]), and that equivalence of single-valued 1NFT is decidable [15]. For two-way transducers, Gurari showed that equivalence is PSPACE-complete in the deterministic case [56], by reducing it to emptiness of bounded-reversal counter machines [59], a polynomial-time solvable problem [57]. Interestingly, the complexity remains the same for single-valued 2NFT, even though, to the best of our knowledge there is no reference available for this latter result.

The above decidability and complexity results can be presented in a uniform way using reductions to equivalence of finite-state automata and to single-valuedness of relational transducers (see Section 3):

▶ **Theorem 11.** *The equivalence problem $T_1 \stackrel{?}{=} T_2$ is*
1. NLOGSPACE*-complete if T_1, T_2 are 1DFT.*
2. PTIME *if T_1, T_2 are unambiguous 1NFT.*
3. PSPACE*-complete if T_1, T_2 are 2DFT, or single-valued 2NFT or NSST.*

Proof. The idea is to reduce the equivalence problem for T_1 and T_2 to testing equivalence of the domains and single-valuedness of the union of T_1 and T_2. For example, when T_1, T_2 are 1DFT, testing equivalence of the underlying deterministic automata is in NLOGSPACE, and the same for testing single-valuedness of the 1NFT $T_1 \cup T_2$ (cf. Theorem 3).

For single-valued 1NFT, it is the complexity of the equivalence problem for the domains that dominates, since this is in general PSPACE-complete [62]. However, if the transducers are unambiguous (which is not a restriction, see e.g. [40, 76]), then $T_1 \stackrel{?}{=} T_2$ can be solved in PTIME [58], since equivalence of unambiguous automata has a polynomial-time solution [78].

Finally, when T_1, T_2 are 2NFT (or NSST), single-valuedness of $T_1 \cup T_2$ is in PSPACE, again by Theorem 3. We need only to show that domain equivalence can also be solved in PSPACE. While this is easy for NSST, it is perhaps not completely obvious that we can check equivalence of two-way, non-deterministic automata in PSPACE. This is indeed the case using e.g. a construction due to Vardi [79]: for every two-way, non-deterministic automaton A of size n, one can construct two deterministic, one-way automata of size $2^{O(n^2)}$, recognizing respectively $L(A)$ and its complement. As usual, PSPACE-hardness for 2NFT follows from emptiness of intersection of finite-state automata [62]; for NSST it follows from universality of finite-state automata [65]. ◀

It is worth noting that the exact complexity for the equivalence problem of DSST is unknown, since the problem is shown to be in PSPACE and NLOGSPACE-hard:

▶ **Open question 12.** *What is the exact complexity of the equivalence problem for DSST?*

A powerful tool that can be used to establish decidability of equivalence problems on transducers is the *Ehrenfeucht conjecture*. It was originally stated as a conjecture about formal languages: for every language $L \subseteq \Sigma^*$, there is a finite subset $F \subseteq L$ such that for every homomorphisms $f, g : \Sigma^* \to \Delta^*$,

$$\forall u \in L \quad f(u) = g(u) \qquad \text{if and only if} \qquad \forall u \in F \quad f(u) = g(u).$$

Such a set F is called a *test set* for L. There is an equivalent formulation of Ehrenfeucht conjecture in terms of word equations [61]. Let Σ and Ω be two alphabets, where the elements in Ω are variables. A word equation is a pair $(u, v) \in \Omega^* \times \Omega^*$, and a solution is a homomorphism $\sigma : \Omega \to \Sigma^*$ such that $\sigma(u) = \sigma(v)$. The Ehrenfeucht conjecture then says that any system of equations over Ω has a finite, equivalent subsystem, where equivalence means that the solution sets are the same. The proof of Ehrenfeucht conjecture is based on encodings of the free monoid and Hilbert's basis theorem [2, 55].

An elegant application of Ehrenfeucht conjecture is due to Culik and Karhumäki, who established decidability of the equivalence problem for k-valued 1NFT [29]. Their decidability result does not come with any complexity upper bound, however the following stronger result leads to an algorithm of elementary complexity:

▶ **Theorem 13** ([80, 72])**.** *Every k-valued 1NFT can be effectively decomposed into a union of k unambiguous 1NFT of exponential size.*

Since k-valuedness of 1NFT can be checked in PTIME [58], this yields:

▶ **Corollary 14.** *The equivalence problem for k-valued 1NFT is in* EXPTIME *(for fixed k).*

Let us come back to the argument used by Culik and Karhumäki for the equivalence of k-valued 1NFT [29]. The proof consists of two parts. First the Ehrenfeucht conjecture is used to show that there is a finite test set for any two k-valued transducers with at most n states (for any fixed n). In this case, a test set is a language F such that *any* two transducers with at most n states are equivalent if and only if they are equivalent on inputs from F. In the second part it is shown how to find effectively a finite test set for the class of k-valued transducers with at most n states. This step amounts to determine whether two finite systems of equations are equivalent, which reduces to solvability of word equations, and can be solved by Makanin's algorithm (see e.g. [37]). This proof idea extends to k-valued 2NFT:

▶ **Theorem 15** ([29])**.** *The equivalence problem for k-valued 2NFT is decidable.*

Proof. We sketch the proof of [29] for one-way transducers, then explain how it adapts to two-way transducers. We tacitly assume that all transducers are trimmed.

For the existence of a test set one uses the formulation of Ehrenfeucht conjecture with word equations. As mentioned, the set we are looking for will be a test set for *any* pair of k-valued transducers with at most n states (and fixed alphabets). An important observation is that for k-valued transducers (one-way or two-way), the following holds: for every pair of states p, q and input letter a, at most k different output words can label the transition (p, a, q). Thanks to this, any k-valued transducer with at most n states can be described by a partial mapping $\Delta : \{1, \dots, n\}^2 \times \Sigma \times \{1, \dots, k\} \to \Omega$ into a *finite* set Ω of variables (a *schema*), paired with an interpretation $\sigma : \Omega \to \Gamma^*$. The (uninterpreted) output of a run that respects a schema Δ is then a word over Ω. For instance, below we depict a transducer (on the left) with the corresponding schema (on the right):

Note that the transducer is 3-valued, and its schema satisfies for instance the equation $x_1 x_3 x_5^m = x_2 x_3 x_5^m$, relating two runs with the same output and over the input $u = a^{m+2}$.

Now, for every pair of concrete k-valued transducers T_1, T_2 with at most n states, and for every input u, the runs of T_1 and respectively T_2, over u, can be partitioned into at most k groups, each associated with an output among the k possible ones. For each group of runs, we can write word equations over Ω as expected. So two transducers as above, being interpretations of some schemata Δ_1, Δ_2, are equivalent only if they satisfy a formula φ_n of the form $\bigwedge_{u \in \Sigma^*} \bigvee_\pi S_\pi$, where π ranges over the possible Δ_1, Δ_2 and partitions the set of runs of Δ_1 and respectively Δ_2, over the input u, into at most k groups, and S_π is the associated system of word equations. Ehrenfeucht conjecture is used in [29] to show that any φ_n as above is equivalent to some *finite* formula $\varphi_{n,m} = \bigwedge_{u \in \Sigma^{\le m}} \bigvee_\pi S_\pi$.

For the effectiveness part, let us assume that for some m, the formulas $\varphi_{n,m}$ and $\varphi_{n,m+1}$ are equivalent, i.e., they have the same sets of solutions (recall that testing the latter equivalence reduces to solvability of word equations). The goal is to prove that $\Sigma^{\le m}$ is a test set, namely, for all $r > m$ and all k-valued transducers T_1, T_2 of size at most n,

$$T_1 \equiv_r T_2 \qquad \text{if and only if} \qquad T_1 \equiv_m T_2$$

where $T_1 \equiv_m T_2$ stands for equivalence relativized to inputs of length at most m. The above property is proved by induction on r, as follows. Suppose for the moment that, given any

transducer T and any input letter a, one can construct a transducer T_a with the same number of states as T and such that $T_a(u) = T(au)$. Clearly, $T_1 \equiv_{r+1} T_2$ is equivalent to $T_{1,a} \equiv_r T_{2,a}$ for every $a \in \Sigma$, and $T_1 \equiv_0 T_2$ (the latter being abbreviated as $(*)$ below). Therefore,

$$\begin{array}{ccccc} T_1 \equiv_{r+1} T_2 & \Leftrightarrow & T_{1,a} \equiv_r T_{2,a} \ (\forall a \in \Sigma) \text{ and } (*) & & T_1 \equiv_m T_2. \\ & & \Updownarrow \text{ (ind. hyp.)} & & \Updownarrow \\ & & T_{1,a} \equiv_m T_{2,a} \ (\forall a \in \Sigma) \text{ and } (*) & \Leftrightarrow & T_1 \equiv_{m+1} T_2 \end{array}$$

This shows that $\Sigma^{\leq m}$ is a test set. According to the first part, we are guaranteed to find such an m.

If T is a 1NFT, one can build T_a while preserving the number of states, just by a shortcut of the transitions departing from the initial state, which can be assumed to have no incoming transition. If T is a 2NFT, a similar idea works, but now the shortcut is more complex since the first input position can be read several times. Still we can shortcut the transitions involving the first input position, assuming, w.l.o.g., that all our transducers have the extra possibility to check whether the current position is the first one. ◀

Recall that single-valued NSST and 2NFT are equivalent transducer models (actually equivalent to DSST and 2DFT). It is natural to ask whether this equivalence extends to k-valued transducers. One direction is an easy generalization of the deterministic case:

▶ **Proposition 16.** *For every k-valued 2NFT, there is an equivalent k-valued NSST.*

▶ **Open question 17.** *Are NSST strictly more expressive than 2NFT in the k-valued case?*

The approach of Culik and Karhumäki [29] does not appear to generalize to NSST (and not even to the equivalent model of 2NFT with common guess, cf. Section 2). The main difficulty is that it is not clear how to construct the transducer T_a from T while preserving the number of states. This difficulty, however, can be overcome for k-valued, 1-register NSST, which thus turn out to have a decidable equivalence problem:

▶ **Theorem 18.** *The equivalence problem for k-valued, 1-register NSST is decidable.*

The above result also follows from [53], where, in analogy with Theorem 13, it is shown that k-valued, 1-register NSST can be effectively decomposed into k unambiguous NSST. Not surprisingly, a generalization of the decomposition theorem for k-valued NSST with multiple registers faces the same difficulties as the approach of Culik and Karhumäki. We conjecture however that Theorem 18 generalizes to concatenation-free NSST:

▶ **Conjecture 19.** *The equivalence problem for k-valued, concatenation-free NSST is decidable.*

The difficulty in proving this conjecture is combinatorial, since it requires to determine if for two partial runs of an NSST, there is some extension that makes their outputs equal.

We conclude this section by presenting a different approach to show equivalence of transducers, that was recently applied to copyful DSST:

▶ **Theorem 20** ([51])**.** *The equivalence problem for copyful DSST is decidable.*

Proof. The original proof of [51] shows that copyful DSST are equivalent to HDT0L systems, and then applies [29]. An alternative proof was presented in [13], translating copyful DSST into so-called polynomial automata, and showing that the zeroness problem for such automata

is decidable. Recently, Bojańczyk reformulated the proof in terms of polynomial grammars [18]. We briefly sketch his proof below.

The starting idea is to encode words by values computed by polynomials. Such encodings were known even before Matiyasevich' negative solution to Hilbert's 10th problem and Makanin's algorithm for word equations. For example, [18] represents a binary word w by the pair $enc(w) = (bin(w), 2^{|w|})$, where $bin(w)$ is the integer with binary representation w. Word concatenation becomes then a polynomial operation: if w_i is encoded by $A_i = (A_{i,1}, A_{i,2})$, then $w_1 w_2$ is encoded by the pair of integers $A_1 \odot A_2 := (A_{1,1}A_{2,2} + A_{2,1},\ A_{1,2}A_{2,2})$. A copyful DSST easily translates into a context-free grammar that generates pairs of integers, a so-called polynomial grammar [18].

For example, for the DSST of Figure 3, the grammar has one non-terminal $A = (A_x, A_y)$ for the unique state, plus a starting symbol S for the final output. The rules are as follows:

$$\begin{array}{rcll} S & \to & A_x \odot enc(\texttt{,}\,\sqcup) \odot A_y & \\ (A_x, A_y) & \to & (enc(\varepsilon),\ A_y \odot enc(\sqcup) \odot A_x) & \\ (A_x, A_y) & \to & (A_x \odot enc(c),\ A_y) & (\forall c \neq \sqcup). \end{array}$$

The question of whether two copyful DSST T, T' are equivalent reduces to asking whether the difference $S - S'$ of the starting symbols of the associated grammars generates only the null vector. Using the decidability of the first-order of reals, there is a semi-algorithm to know if a polynomial grammar can generate some non-zero value, by enumerating derivations. Using Hilbert's basis theorem, there is also a semi-algorithm for proving that a polynomial grammar generates only the null vector, by enumerating finite sets of polynomials and checking that (1) they induce a solution for the grammar, and (2) they produce only zero (see [18] for details). This shows that one can decide whether a polynomial grammar generates only the null vector, hence the equivalence of copyful DSST. ◀

5 Transducers with origins

Transducers with origin information have been introduced by Bojańczyk in [17] as a way to recover the algebraic world that appears to get lost when switching from regular string languages to regular string transductions. We recall that for 2DFT or DSST there is no canonical (e.g. minimal) model on which properties like aperiodicity can be tested. As we saw in Section 3, the situation is slightly better for one-way transducers, that do have canonical models – the minimal transducer in the deterministic case, and the canonical bimachine in the general case.

In the standard semantics, a string transduction is simply a relation $R \subseteq \Sigma^* \times \Gamma^*$ that consists of pairs of words. In the origin semantics, a transduction is a set of pairs from $\Sigma^* \times (\Gamma \times \mathbb{N})^*$ such that the output from $(\Gamma \times \mathbb{N})^*$ also records at which position of the input it was generated. Every formalism used for describing transductions, be it transducers, logic or expressions, can be enriched by the origin information in a natural way. For example, the origin semantics for 1NFT is the same as a synchronization language – recall from Section 2 that this is a language of interleavings of input and output letters, and that there could be many synchronization languages representing the same relation. For 2NFT, the origin semantics is conveniently described by *origin graphs*, which are special graphs consisting of two total orders (for the input and the output), together with edges from output positions to input positions (the origin mapping). For example, the two origin graphs of Figure 4 are different, although they have the same input/output pair.

Recall that a string-to-string function is called regular if it is realized by one of the following, equivalent formalisms: 2DFT, DSST, MSO transductions. The main contribution

Figure 4 Two origin graphs; the upper line is the input, the lower line the output.

of [17] was to show a Myhill-Nerode theorem for regular functions with origin semantics, and give an effective characterization for the subclass of FO transductions. The result amounts to define left and right congruences, as follows. Recall that the Nerode (right) congruence for a language K defines $u \equiv v$ if $u^{-1}K = v^{-1}K$. Similarly, for a function $f : \Sigma^* \to \Gamma^*$, we define a right congruence by $u \equiv_R v$ if $f_{u\_} = f_{v\_}$, where $f_{u\_}(w)$ is obtained from $f(uw)$ by replacing all maximal non-empty factors with origins inside u by a special marker $\bullet$. For example, for $f(w) = ww$ there are two possibilities for $f_{u\_}$, depending on whether u is empty or not: one is $v \mapsto vv$, and the other is $v \mapsto \bullet\, v \bullet v$. Symmetrically, one defines a left congruence $\equiv_L$ using $f_{\_u}(w)$, that is obtained from $f(wu)$ by replacing with $\bullet$ all maximal non-empty factors with origins inside u.

▶ **Theorem 21** ([17]). *Let f be a string-to-string function with origins.*

1. *f is regular if and only if the associated congruences $\equiv_L$ and $\equiv_R$ have finite index.*
2. *A regular f is origin-equivalent to an FO transduction if and only if all classes of the associated congruences $\equiv_L$ and $\equiv_R$ are FO-definable languages.*

An immediate consequence of the above theorem is that one can decide FO-definability in the origin semantics. Using that the congruences $\equiv_L$ and $\equiv_R$ can be tested in Pspace we get:

▶ **Corollary 22.** *The problem whether given 2DFT or DSST is origin-equivalent to some FO transduction is* Pspace*-complete.*

Register minimization of DSST is another problem that can be solved under the origin semantics, as stated below. The main ingredient of the characterization is the notion of *k-crossing*. Formally, an output position y is said to cross an input position x if the origin of y is x or to the left of x, and the origin of $y + 1$ is to the right of x (if $y + 1$ is defined). An origin graph is k-crossing if every input position is crossed by at most k output positions.

▶ **Theorem 23** ([19]). *A set $\mathcal{G}$ of origin graphs is realizable by a k-register DSST if and only if it satisfies all conditions below:*

1. *every graph in $\mathcal{G}$ is k-crossing,*
2. *there is a constant c such that, for all graphs in $\mathcal{G}$, every input position is the origin of at most c output positions,*
3. *$\mathcal{G}$ is MSO-definable.*

Taking the right hand-side graph of Figure 4 as an example, one sees that $c = k = 1$, and the MSO formula defining such graphs says, among other things, that the origin of the output position $y + 1$ is the predecessor of the origin of y.

Another quite interesting phenomenon is that the equivalence problem becomes decidable in the origin semantics, even for non-deterministic, two-way transducers with common guess:

▶ **Theorem 24** ([21]). *The origin-equivalence problem for 2NFT (possibly enhanced with common guess) is* Pspace*-complete.*

The previous result is consistent with the intuition that under the origin semantics, one does not reason anymore on two separate objects (input and output), but on one single object (the origin graph). In this perspective, it becomes natural to ask whether some amount of information can be removed from the origin semantics, while still preserving decidability of equivalence. The argument below reveals that not much information can be removed.

Consider a modified notion of origin, that instead of mapping output positions to input positions, defines an equivalence on the output so that any two positions are equivalent when they have the same origin in the input. A small modification of the proof of Theorem 2 show that decidability of equivalence is lost under this weaker semantics. Indeed, one can replace the pairs (wu, c^n) that encode non-solutions of the PCP instance by pairs of the form $(w'u', c^n)$, where $w'u'$ is obtained from wu by inserting a fixed dummy word $\sqcup \cdots \sqcup$ between every two consecutive positions, so that it becomes possible to produce at most one output letter at each input position. In this case, the equal-origin information becomes vacuous, and the universality problem turns out to be undecidable.

6 Conclusions

We have reviewed some of the fundamental questions on string transducers, mostly concerning characterization and equivalence problems. Some of the problems were shown decidable, notably in the single-valued and in the finite-valued case, but a few important problems remain open.

This is the case, for instance, for the FO-definability problem for regular functions: given a 2DFT or a DSST, is it equivalent to some FO transduction? As a possible first attempt, one may try to look at FO-definability for restricted variants of 2DFT and DSST (e.g. sweeping 2DFT and concatenation-free DSST), trying to lift the known characterization for 1DFT.

Another challenging question that remains unanswered is whether k-valued NSST are as expressive as k-valued 2NFT, or strictly more expressive. The related problem of testing equivalence of k-valued NSST is also open. A solution to both problems might stem from a decomposition theorem, that is, from a proof that every k-valued NSST can be decomposed into an equivalent finite union of DSST. This latter result however seems rather difficult to get, due to the underlying word combinatorics. Also in this case, it might be helpful to consider first the problem for the restricted class of concatenation-free NSST.

We finally recall a couple of open problems related to equivalence and single-valuedness of SST. We have seen that equivalence for single-valued NSST is PSPACE-complete, but what about DSST? Is the equivalence problem still PSPACE-hard, or is it possible to exploit determinism to lower the complexity? Similarly, the problem of testing singe-valuedness on NSST is shown to be in PSPACE, but no matching lower bound is known. Finally, the problem of minimizing the number of registers in a DSST is also open.

References

1 Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. A general theory of translation. *Math. Syst. Theory*, 3(3):193–221, 1969.

2 M.H. Albert and J. Lawrence. A proof of Ehrenfeucht's conjecture. *Theor. Comput. Sci.*, 41(1):121–123, 1985.

3 Rajeev Alur and Pavel Cerný. Expressiveness of streaming string transducer. In *IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS'10)*, volume 8 of *LIPIcs*, pages 1–12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010.

4 Rajeev Alur and Jyotirmoy Deshmukh. Nondeterministic streaming string transducers. In *International Colloquium on Automata, Languages and Programming (ICALP'11)*, volume 6756 of *LNCS*. Springer, 2011.

5 Rajeev Alur, Emmanuel Filiot, and Ashutosh Trivedi. Regular Transformations of Infinite Strings. In *Proc. of Annual IEEE Symposium on Logic in Computer Science (LICS'12)*, pages 65–74. IEEE, 2012.

6 Rajeev Alur, Adam Freilich, and Mukund Raghothaman. Regular combinators for string transformations. In *Joint meeting of Annual Conference on Computer Science Logic (CSL) and Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, LIPIcs, pages 9:1–9:10. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2014.

7 Félix Baschenis, Olivier Gauwin, Anca Muscholl, and Gabriele Puppis. One-way Definability of Sweeping Transducers. In *IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS'15)*, volume 45 of *LIPIcs*, pages 178–191. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015.

8 Félix Baschenis, Olivier Gauwin, Anca Muscholl, and Gabriele Puppis. Minimizing Resources of Sweeping and Streaming String Transducers. In *International Colloquium on Automata, Languages, and Programming (ICALP'16)*, volume 55 of *LIPIcs*, pages 114:1–114:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.

9 Félix Baschenis, Olivier Gauwin, Anca Muscholl, and Gabriele Puppis. Untwisting two-way transducers in elementary time. In *ACM/IEEE Symposium on Logic in Computer Science (LICS'17)*. IEEE Computer Society, 2017.

10 Félix Baschenis, Olivier Gauwin, Anca Muscholl, and Gabriele Puppis. One-way definability of two-way word transducers. *Logical Methods in Computer Science*, 14(4):1–54, 2018.

11 Marie-Pierre Béal and Olivier Carton. Determinization of transducers over finite and infinite words. *Theor. Comput. Sci.*, 289(1):225–251, 2002.

12 Marie-Pierre Béal, Olivier Carton, Christophe Prieur, and Jacques Sakarovitch. Squaring transducers: an efficient procedure for deciding functionality and sequentiality. *Theor. Comput. Sci.*, 292:45–63, 2003.

13 Michael Benedikt, Timothy Duff, Aditya Sharad, and James Worrell. Polynomial automata: Zeroness and applications. In *Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'17)*, pages 1–12. IEEE, 2017.

14 Jean Berstel. *Transductions and context-free languages*. Teubner Studienbücher Stuttgart, 1979.

15 Meera Blattner and Tom Head. Single-valued a-transducers. *J. Comput. and System Sci.*, 15:310–327, 1977.

16 Meera Blattner and Tom Head. The decidability of equivalence for deterministic finite transducers. *J. Comput. and System Sci.*, 19:45–49, 1979.

17 Mikolaj Bojańczyk. Transducers with origin information. In *International Colloquium on Automata, Languages and Programming (ICALP'14)*, number 8572 in LNCS, pages 26–37. Springer, 2014.

18 Mikolaj Bojańczyk. The Hilbert Method for Transducer Equivalence. *ACM SIGLOG News*, January 2019.

19 Mikolaj Bojańczyk, Laure Daviaud, Bruno Guillon, and Vincent Penelle. Which Classes of Origin Graphs Are Generated by Transducers? In *International Colloquium on Automata, Languages and Programming (ICALP'17)*, volume 80 of *LIPIcs*, pages 114:1–114:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

20 Mikolaj Bojańczyk, Laure Daviaud, and Shankara Narayanan Krishna:. Regular and First-Order List Functions. In *Proc. of Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2018)*, pages 125–134. ACM, 2018.

21 Sougata Bose, Anca Muscholl, Vincent Penelle, and Gabriele Puppis. Origin-Equivalence of Two-Way Word Transducers Is in PSPACE. In *IARCS Annual Conference on Foundations of*

Software Technology and Theoretical Computer Science (FSTTCS'18), volume 122 of *LIPIcs*, pages 1–18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

22 Julius Richard Büchi. Weak Second-order Arithmetic and Finite Automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960.

23 Olivier Carton, Christian Choffrut, and Serge Grigorieff. Decision problems among the main subfamilies of rational relations. *ITA*, 40(2):255–275, 2006.

24 Olivier Carton and Luc Dartois. Aperiodic two-way transducers and FO-transductions. In *Proc. of EACSL Annual Conference on Computer Science Logic (CSL'15)*, LIPIcs, pages 160–174. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015.

25 Olivier Carton, Léo Exibard, and Olivier Serre. Two-Way Two-Tape Automata. In *Proc. in Developments in Language Theory (DLT'17)*, number 10396 in LNCS, pages 147–159. Springer, 2017.

26 Christian Choffrut. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theor. Comput. Sci.*, 5:325–338, 1977.

27 Christian Choffrut. Minimizing subsequential transducers: a survey. *Theor. Comput. Sci.*, 292(131-143), 2003.

28 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic – A Language-Theoretic Approach*, volume 138 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2012.

29 Karel Culik II and Juhani Karhumäki. The equivalence of finite valued transducers (on HDT0L languages) is decidable. *Theor. Comput. Sci.*, 47:71–84, 1986.

30 Karel Culik II and Juhani Karhumäki. The Equivalence Problem for Single-Valued Two-Way Transducers (on NPDT0L Languages) is Decidable. *SIAM J. Comput*, 16(2):221–230, 1987.

31 Luc Dartois, Emmanuel Filiot, Pierre-Alain Reynier, and Jean-Marc Talbot. Two-Way Visibly Pushdown Automata and Transducers. In *Proc. of Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'16)*, pages 217–226. ACM, 2016.

32 Luc Dartois, Paulin Fournier, Ismaël Jecker, and Nathan Lhote. On Reversible Transducers. In *Proc. of International Colloquium on Automata, Languages, and Programming (ICALP'17,)*, number 113 in LIPIcs, pages 1–12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

33 Vrunda Dave, Paul Gastin, and Shankara Narayanan Krishna. Regular Transducer Expressions for Regular Transformations. In *Proc. of Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2018)*, pages 315–324. ACM, 2018.

34 Laure Daviaud, Pierre-Alain Reynier, and Jean-Marc Talbot. A Generalised Twinning Property for Minimisation of Cost Register Automata. In *Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '16)*, pages 857–866. ACM, 2016.

35 María Emilia Descotte, Diego Figueira, and Santiago Figueira. Closure properties of synchronized relations. In *International Symposium on Theoretical Aspects of Computer Science (STACS'19)*, LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. to appear.

36 María Emilia Descotte, Diego Figueira, and Gabriele Puppis. Resynchronizing Classes of Word Relations. In *International Colloquium on Automata, Languages, and Programming (ICALP'18)*, volume 123 of *LIPIcs*, pages 1–13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

37 Volker Diekert. Makanin's Algorithm. In M. Lothaire, editor, *Algebraic combinatorics on words*. Cambridge University Press, 2002.

38 Volker Diekert, Paul Gastin, and Manfred Kufleitner. A survey on Small Fragments of First-Order Logic over Finite Words. *International Journal of Foundations of Computer Science*, 19(3):513–548, 2008.

39 Volker Diekert and Manfred Kufleitner. A Survey on the Local Divisor Technique. *Theor. Comput. Sci.*, 610:13–23, 2016.

40 Samuel Eilenberg. *Automata, Languages and Machines*. Academic Press, New York, 1976.

41 Calvin C. Elgot and Jorge E. Mezei. On Relations Defined by Generalized Finite Automata. *IBM Journal of Research and Development*, 9(1):47–68, 1965.

42 Joost Engelfriet and Hendrik Jan Hoogeboom. MSO definable string transductions and two-way finite-state transducers. *ACM Trans. Comput. Log.*, 2(2):216–254, 2001.

43 Diego Figueira and Leonid Libkin. Synchronizing Relations on Words. *Theory Comput. Syst.*, 57(2):287–318, 2015.

44 Emmanuel Filiot, Olivier Gauwin, and Nathan Lhote. Aperiodicity of Rational Functions Is PSPACE-Complete. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'16)*, volume 65 of *LIPIcs*, pages 13:1–13:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.

45 Emmanuel Filiot, Olivier Gauwin, Nathan Lhote, and Anca Muscholl. On Canonical Models for Rational Functions over Infinite Words. In *Proc. of IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'18)*, volume 30 of *LIPIcs*, pages 1–17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

46 Emmanuel Filiot, Olivier Gauwin, Pierre-Alain Reynier, and Frédéric Servais. From Two-Way to One-Way Finite State Transducers. In *ACM/IEEE Symposium on Logic in Computer Science (LICS'13)*, pages 468–477, 2013.

47 Emmanuel Filiot, Ismaël Jecker, Christof Löding, and Sarah Winter. On Equivalence and Uniformisation Problems for Finite Transducers. In *Proc. of nternational Colloquium on Automata, Languages, and Programming (ICALP'16)*, number 125 in LIPIcs, pages 1–14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.

48 Emmanuel Filiot, Shankara Narayanan Krishna, and Ashutosh Trivedi. First-order Definable String Transformations. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'14)*, LIPIcs, pages 147–159. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2014.

49 Emmanuel Filiot, Jean-François Raskin, Pierre-Alain Reynier, Frédéric Servais, and Jean-Marc Talbot. Visibly pushdown transducers. *J. Comput. Syst. Sci.*, 97:147–181, 2018.

50 Emmanuel Filiot and Pierre-Alain Reynier. Transducers, logic and algebra for functions of finite words. *ACM SIGLOG News*, pages 4–19, 2016.

51 Emmanuel Filiot and Pierre-Alain Reynier. Copyful Streaming String Transducers. In *International Workshop on Reachability Problems (RP'17)*, number 10506 in LNCS, pages 75–86. Springer, 2017.

52 Patrick C. Fischer and Arnold L. Rosenberg. Multi-tape one-way nonwriting automata. *J. Comput. and System Sci.*, 2:88–101, 1968.

53 Paul Gallot, Anca Muscholl, Gabriele Puppis, and Sylvain Salvati. On the Decomposition of Finite-Valued Streaming String Transducers. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS'17)*, volume 66 of *LIPIcs*, pages 34:1–34:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

54 Seymour Ginsburg and Gene F. Rose. A characterization of machine mappings. *Canad. J. Math.*, 18:381–388, 1966.

55 Victor S. Guba. Equivalence of infinite systems of equations in free groups and semigroups to finite subsystems. *Mat. Zametki*, 40(3):688—-690, 1986.

56 Eitan M. Gurari. The equivalence problem for deterministic two-way sequential transducers is decidable. *SIAM Journal of Computing*, 448–452, 1982.

57 Eitan M. Gurari and Oscar H. Ibarra. The complexity of decision problems for finite-turn multicounter machines. *J. Comput. and System Sci.*, 16(1):61–66, 1981.

58 Eitan M. Gurari and Oscar H. Ibarra. A note on finite-valued and finitely ambiguous transducers. *Math. Syst. Theory*, 16(1):61–66, 1983.

59 Oscar H. Ibarra. Reversal-bounded multicounter machines and their decision problems. *JACM*, 1978.

60 Oscar H. Ibarra. The unsolvability of the equivalence problem for e-free NGSM's with unary input (output) alphabet and applications. *SIAM J. of Comput.*, 7(4):524–532, 1978.

61 Juhani Karhumäki. The Ehrenfeucht conjecture: a compactness claim for finitely generated free monoids. *Theor. Comput. Sci.*, 29:285–308, 1984.

62 Dexter Kozen. Lower bounds for natural proof systems. In *Annual Symposium on Foundations of Computer Science (FOCS'77)*, pages 254–266. IEEE, 1977.

63 Christof Löding and Christopher Spinrath. Decision Problems for Subclasses of Rational Relations over Finite and Infinite Words. *Discrete mathematics and theoretical computer science*, 2019.

64 Robert McNaughton and Seymour Papert. *Counter-Free Automata*. MIT Press, 1971.

65 Albert R. Meyer and Larry J. Stockmeyer. The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space. In *13th Annual Symposium on Switching and Automata Theory*, pages 125–129. IEEE Computer Society, 1972.

66 Edward F Moore. Gedanken-experiments on sequential machines. *Automata studies*, 34:129–153, 1956.

67 Maurice Nivat. Transduction des langages de Chomsky. *Annales de l'Institut Fourier*, 18:339–455, 1968.

68 Jean-Eric Pin. Logic, Semigroups and Automata on Words. *Annals of Mathematics and Artificial Intelligence*, 16:343–384, 1996.

69 Michael O. Rabin and Dana Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, pages 114–125, 1959.

70 Christophe Reutenauer and Marcel-Paul Schützenberger. Minimization of rational word functions. *SIAM Journal of Computing*, 20(4):669–685, 1991.

71 Sasha Rubin. Automata Presenting Structures: A Survey of the Finite String Case. *Bulletin of Symbolic Logic*, 14(2):169–209, 2008.

72 Jacques Sakarovitch and Rodrigo de Souza. Lexicographic decomposition of K-valued transducers. *Theory Comput. Sci.*, 47:758–785, 2010.

73 Marcel-Paul Schützenberger. A remark on finite transducers. *Information and Control*, 4(2-3):185–196, 1961.

74 Marcel-Paul Schützenberger. On Finite Monoids Having Only Trivial Subgroups. *Information and Control*, 8:190–194, 1965.

75 Marcel-Paul Schützenberger. Sur les relations rationnelles. In *Proc. of 2nd GI conference, Automata Theory and Formal Languages*, number 33 in LNCS, pages 209–213. Springer, 1975.

76 Marcel-Paul Schützenberger. Sur les relations rationnelles entre monoïdes libres. *Theor. Comput. Sci.*, 3(2):243–259, 1976.

77 John C. Shepherdson. The reduction of two-way automata to one-way automata. *IBM Journal of Research and Development*, 3(2):198–200, 1959.

78 Richard E. Stearns and Harry B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, grammars and automata. In *Annual Symposium on Foundations of Computer Science (FOCS'81)*, pages 74–81, 1981.

79 Moshe Y. Vardi. A Note on the Reduction of Two-Way Automata to One-Way Automata. *Information Processing Letters*, 30:261–264, 1989.

80 Andreas Weber. Decomposing A k-Valued Transducer into k Unambiguous Ones. *RAIRO-ITA*, 30(5):379–413, 1996.

81 Andreas Weber and Reinhard Klemm. Economy of description for single-valued transducers. *Inf. Comput.*, pages 327–340, 1995.

82 Thomas Wilke. Classifying Discrete Temporal Properties. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS'99)*, volume 1563 of *LNCS*, pages 32–46. Springer, 1999.