

# Lexicographic Transductions of Finite Words

Emmanuel Filiot ✉ 

Université libre de Bruxelles (ULB), Belgium

Nathan Lhote ✉ 

Aix Marseille Univ, CNRS, LIS, Marseille, France

Pierre-Alain Reynier ✉

Aix Marseille Univ, CNRS, LIS, Marseille, France

---

## Abstract

Regular transductions over finite words have linear input-to-output growth. This class of transductions enjoys many characterizations, such as transductions computable by two-way transducers as well as transductions definable in MSO (in the sense of Courcelle). Recently, regular transductions have been extended by Bojańczyk to polyregular transductions, which have polynomial growth, and are characterized by pebble transducers and MSO interpretations. Another class of interest is that of transductions defined by streaming string transducers or marble transducers, which have exponential growth and are incomparable with polyregular transductions.

In this paper, we consider MSO set interpretations (MSOSI) over finite words, that were introduced by Colcombet and Loeding. MSOSI are a natural candidate for the class of “regular transductions with exponential growth”, and are rather well behaved. However, MSOSI for now lacks two desirable properties that regular and polyregular transductions have. The first property is to have an automata description. This property is closely related to a second property, that of being regularity preserving, meaning preserving regular languages under inverse image.

We first show that if MSOSI are (effectively) regularity preserving then any automatic  $\omega$ -word has a decidable MSO theory, an almost 20 years old conjecture of Bárány.

Our main contribution is the introduction of a class of transductions of exponential growth, which we call lexicographic transductions. We provide three different presentations for this class: first, as the closure of simple transductions (recognizable transductions) under a single operator called maplex; second, as a syntactic fragment of MSOSI (but the regular languages are given by automata instead of formulas); and third, we give an automaton based model called nested marble transducers, which generalize both marble transducers and pebble transducers. We show that this class enjoys many nice properties including being regularity preserving.

**2012 ACM Subject Classification** Theory of computation → Automata extensions

**Keywords and phrases** Transducers, Automata, MSO, Logical interpretations, Automatic structures

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2025.50

**Related Version** *Full Version:* <https://arxiv.org/abs/2503.01746> [18]

**Funding** This work has been partly funded by the QuaSy project (ANR-23-CE48-0008).

*Emmanuel Filiot:* research director at F.R.S.-FNRS. This work was partially funded by the FNRS project 40020726.

**Acknowledgements** Nathan Lhote would like to warmly thank Mikołaj Bojańczyk, Rafał Stefański and Lê Thành Dũng (Tito) Nguyễn for many interesting discussions at different stages of this work.

## 1 Introduction

Before discussing lexicographic transductions, the central notion of this article, we give some context on transduction classes and connections to automatic structures.



© Emmanuel Filiot, Nathan Lhote, and Pierre-Alain Reynier;  
licensed under Creative Commons License CC-BY 4.0

50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025).

Editors: Paweł Gawrychowski, Filip Mazowiecki, and Michał Skrzypczak; Article No. 50; pp. 50:1–50:18



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

**MSOSI and the connection to automatic structures.** MSO set interpretations (MSOSI) were introduced in [11], as a generalization of automatic structures (as well as  $\omega$ -automatic, tree automatic and  $\omega$ -tree automatic structures). Indeed an automatic structure can be seen as an MSOSI whose domain is a single structure with decidable MSO theory such as  $(\mathbb{N}, \leq)$ . Using a framework of transformations turns out to be very fruitful, and most of the properties of automatic structures already hold for set-interpretations over structures with decidable MSO theory. The core property of automatic structures (and their generalizations) is that they have a decidable FO theory. More generally, MSOSI have what we call the *FO backward translation property*, meaning that the inverse image of an FO definable set by an MSOSI is MSO definable. This property is obtained via simple, yet powerful, syntactic formula substitution. This technique actually allows to show more generally that MSOSI are closed under post-composition by FO interpretations (FOI).

Generally speaking, automatic structures do not have a decidable MSO theory. This has motivated a line of research that looks for interesting structures with a decidable MSO theory. For instance, morphic  $\omega$ -words, as well as two generalizations called  $k$ -lexicographic  $\omega$ -words [2] and toric  $\omega$ -words [4], have been shown to have a decidable MSO theory. Morphic  $\omega$ -words and  $k$ -lexicographic  $\omega$ -words are particular cases of *automatic  $\omega$ -words*<sup>1</sup>. An automatic  $\omega$ -word is an automatic structure with unary relations and a single binary relation that is a total order isomorphic to  $(\mathbb{N}, \leq)$  (it is crucial that the structure be given by its order relation and not by the successor). To the best of our knowledge, it is not known whether an automatic  $\omega$ -word with an undecidable MSO theory exists, raising the next conjecture.

► **Conjecture 1** ([2], Section 9). *Any automatic  $\omega$ -word has a decidable MSO theory*<sup>2</sup>.

In [2, Corollary 5.6], the author even shows that  $k$ -lexicographic  $\omega$ -words are closed under sequential transductions. As we show in Proposition 11 this property is deeply connected to preserving MSO definable sets by inverse image (which we call *regularity preserving*<sup>3</sup>) and is stronger than having a decidable MSO theory.

A different setting where one can obtain regularity preserving transductions, is provided in [8] where it is shown that MSO interpretations (MSOI) from finite words to finite words characterize the polyregular transductions. Once again, as for automatic  $\omega$ -words, the output structure must be defined by its order and not by the successor.

This calls for a more unifying argument and systematic study of MSOSI whose output structures are linearly ordered, that we phrase as a conjecture<sup>4</sup>:

► **Conjecture 2.** *MSOSI from finite words to finite words are regularity preserving.*

In this article, we focus on transductions from finite words to finite words for two main reasons: it is already challenging, and it captures part of the difficulty of  $\omega$ -words. We show in Proposition 14 that a positive answer to Conjecture 2 entails that Conjecture 1 holds.

**On regular transductions with exponential growth.** The theory of finite word transducers has a long history (in fact as long as automata theory) and is still actively studied.

Various classes of transductions have been introduced, most notably (and ordered inclusion-wise): sequential (Seq), rational (Rat), regular (Reg) and the more recent polyregular transductions (PolyReg) as well as transductions defined by streaming string transducers (SST), which subsume Reg but are incomparable to PolyReg. For a recent survey, see [23].

<sup>1</sup> Not to be confused with automatic *sequences*.

<sup>2</sup> Bárány actually conjectures that any automatic  $\omega$ -word has a so-called *canonical presentation*.

<sup>3</sup> It follows the terminology of [3]. This property is sometimes called regular continuity [10].

<sup>4</sup> One could state stronger conjectures extending the structures to trees,  $\omega$ -words or infinite trees.

These classes are rather well-known and enjoy nice *regularity* properties, including being closed under composition (except for SST which is still closed under post-composition by Seq) which entails<sup>5</sup> being regularity preserving. However some important questions remain open, such as equivalence of PolyReg transductions, which is not known to be decidable.

The two classes of Reg and PolyReg enjoy natural logical characterizations, namely word-to-word MSO transductions (MSOT) and MSOI, respectively. The fact that MSOT are regularity preserving is again obtained by simple formula substitution and holds for arbitrary structures. In contrast in the case of MSOI, the only known proof is *via* a translation into an automaton model called pebble transducers. This raises a natural question for MSOI:

► **Question 1.** *Can we get an automaton model corresponding to MSOI over finite words?*

A positive answer to this question would hopefully provide a proof of Conjecture 2, since natural automata models are usually closed under post-composition by Seq<sup>5</sup>. While hope plays an important part in research, we have good reasons to think this is a hard problem: as mentioned above this would solve a long standing open problem on automatic structures.

It is rather clear that PolyReg captures the “right” notion of regular transductions with polynomial growth. While MSOI seems like a natural candidate, not enough is known about this class yet to say that it captures the “right” notion of regular transductions with exponential growth. Let us more humbly describe what should be, in our view, a *nice* class of regular transductions with exponential growth: this class should be characterized by different, somewhat natural<sup>6</sup>, computation models which subsume the well-behaved classes of PolyReg and SST. It should be regularity preserving and potentially<sup>5</sup> have extra closure properties by pre- or post-composition with smaller classes. In this article we introduce the class of *lexicographic transductions* (Lex) which meets all the above criteria.

**Contributions.** The first contribution of the article is a hardness result: showing that word-to-word MSOI are regularity preserving is at least as hard as showing that any automatic  $\omega$ -word has a decidable MSO theory (Proposition 14). To obtain this result we define *automatic transduction* (AT) which are naturally equivalent to MSOI but formulated in a way that makes the connection with automatic structures clearer. That way we obtain a one-to-one correspondence between automatic  $\omega$ -words and automatic transductions over a unary alphabet which define a total function.

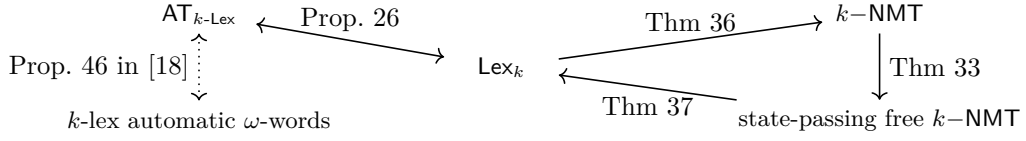
The main contribution of the article is the introduction of a new class of transductions, called *lexicographic transductions* (Lex). We give three different characterizations of this class and show that it enjoys many nice properties, including being regularity preserving.

The first definition of Lex is in the spirit of list functions of [5, 7]: we start with *simple functions* which are recognizable transductions whose range contains words of length at most 1 only. Then we close the class under a single type of operator called **maplex** which works as follows: **maplex**  $f$  maps a word  $u$  to the concatenation  $f(u_1)f(u_2)\dots f(u_n)$  where  $u_1, \dots, u_n$  are all the labellings of  $u$  over some fixed and totally ordered alphabet, enumerated in lexicographic order.

Secondly, we show that this class can be expressed as a syntactic restriction of AT, which we call lexicographic automatic transductions ( $\text{AT}_{\text{Lex}}$ ). These two characterizations are actually syntactically equivalent but quite different in spirit. We leverage the aforementioned correspondence between automatic  $\omega$ -words and automatic transduction, as well as a result of Bárány to show that the nesting of **maplex** operators generates a strict hierarchy of transduction classes (Proposition 27).

<sup>5</sup> In Proposition 11 we see that the two are closely related.

<sup>6</sup> As opposed to an artificial model like the union of PolyReg with SST.



■ **Figure 1** Overview of the different equivalent models, with the transformations between them. The dotted arrow denotes an equivalence with a semantic restriction of  $\text{AT}_{k\text{-Lex}}$  (see [18] for details).

Thirdly, we introduce an automaton model called *nested marble transducers* (NMT). Nested marble transducers are quite expressive: they generalize marble transducers [12, 15] which are known to coincide with SST, they also naturally generalize PolyReg. Informally, a level  $k$  nested marble transducer can annotate its input as a marble transducer (i.e. it drops a marble whenever moving left and lifts a marble whenever moving right), and call a level  $k-1$  nested marble transducer to run on this annotated configuration. This call returns both an output string and a state which the top-level transducer can use to take its next transition. This passing of information from the lower levels to the higher levels is what allows to prove closure under post-composition with a sequential transducer. This is the key ingredient to show that NMT have regular domains, are regularity preserving, and more generally are closed under post-composition by PolyReg.

Regarding expressiveness, Lex can be expressed by NMT in a rather direct way (Theorem 36). In the other way, transductions expressed in Lex do not have such a state-passing mechanism, hence showing that NMT is included in Lex constitutes the technical heart of this article (Theorem 37). An important step consists in showing that one can remove the state-passing mechanism in NMT (Theorem 33). On top of being technical, we show it is computationally costly: there is an unavoidable non-elementary blow-up to transform a nested marble transducers into nested marble transducers without state-passing. This allows to prove the equivalence between the different models. In addition, as shown in Figure 1, this equivalence holds at each level of the aforementioned hierarchy.

## 2 Word languages and transductions

**Words and languages.** Given an alphabet  $\Sigma$ , a  $\Sigma$ -word  $u$  (or just word if  $\Sigma$  is clear from the context) is a sequence of letters from  $\Sigma$ . We denote by  $\epsilon$  the empty word, and by  $|u|$  the length of a word  $u$ . In particular  $|\epsilon| = 0$ . For all integers  $n \geq 0$ , we let  $\Sigma^n$  (resp.  $\Sigma^{\leq n}$ ) be the set of words of length  $n$  (resp. at most  $n$ ). We let  $\text{Pos}(u) = \{1, \dots, |u|\}$  be the set of *positions* of  $u$ , and for all  $i \in \text{Pos}(u)$ ,  $u[i] \in \Sigma$  is the  $i$ -th letter of  $u$ . We write  $\Sigma^*$  for the set of words over  $\Sigma$ , and  $\Sigma^+$  for the set of non-empty words. An  $\omega$ -word is defined similarly, with a set of positions equal to  $\mathbb{N}$ . A word language over  $\Sigma$  is a subset of  $\Sigma^*$ . *In this paper, we let  $|$  be a symbol called separator, assumed to be distinct from any alphabet symbol.* Let  $\Sigma_1, \Sigma_2$  be two alphabets,  $\ell \in \mathbb{N}$  and  $u_1 \in \Sigma_1^\ell, u_2 \in \Sigma_2^\ell$  be two words of length  $\ell$ . The *convolution*  $u_1 \otimes u_2$  of  $u_1$  and  $u_2$  is the word in  $(\Sigma_1 \times \Sigma_2)^\ell$  such that for all  $1 \leq i \leq \ell$ ,  $(u_1 \otimes u_2)[i] = (u_1[i], u_2[i])$ .

**Finite automata.** A (non-deterministic) finite automaton (NFA) over an alphabet  $\Sigma$  is denoted as a tuple  $A = (Q, q_0, F, \Delta)$  where  $Q$  is the set of states,  $q_0$  the initial state,  $F \subseteq Q$  the final states, and  $\Delta \subseteq Q \times \Sigma \times Q$  the transition relation. We write  $q \xrightarrow{u}_A q'$  when there exists a run of  $A$  from state  $q$  to state  $q'$  on  $u$ , and denote by  $L(A) = \{u \in \Sigma^* \mid q_0 \xrightarrow{u}_A q_f \in F\}$  the language recognized by  $A$ . When  $A$  is a deterministic finite automaton (DFA), the transition relation is denoted by a (partial) function  $\delta : Q \times \Sigma \rightarrow Q$ .

**Word transductions.** A *word transduction* (or just transduction for short) over  $\Sigma, \Gamma$  two alphabets is a (partial) function  $f: \Sigma^* \rightarrow \Gamma^*$ . We denote by  $\text{dom}(f)$  its domain. Given two transductions  $f_1, f_2: \Sigma^* \rightarrow \Gamma^*$  with disjoint domains, we let  $f_1 + f_2$  be the transduction of domain  $\text{dom}(f_1) \cup \text{dom}(f_2)$  such that  $(f_1 + f_2)(u) = f_i(u)$  if  $u \in \text{dom}(f_i)$ . Given  $f: \Sigma^* \rightarrow \Gamma^*$ ,  $g: \Gamma^* \rightarrow \Lambda^*$ , we write  $(g f): \Sigma^* \rightarrow \Lambda^*$  the composition  $g \circ f$ . Given  $h: \Lambda^* \rightarrow \Delta^*$ ,  $(h g f)$  stands for  $(h (g f))$ . For  $u \in \Sigma^*$ , we also write  $(h g f u)$  for  $(h g f)(u)$ .

A transduction  $f$  has *exponential growth* if there exists  $c \in \mathbb{N}$  such that for all  $u \in \text{dom}(f)$ ,  $|f(u)| \leq 2^{c|u|}$  holds. A transduction  $f$  has *polynomial growth* if there exist  $c, k \in \mathbb{N}$  such that for all  $u \in \text{dom}(f)$ ,  $|f(u)| \leq c|u|^k$  holds.

► **Example 3** (Reverse, copy and square). Let  $\Sigma$  be an alphabet. The transduction  $\text{rev}: \Sigma^* \rightarrow \Sigma^*$  takes as input any word  $u = \sigma_1 \dots \sigma_n$  and outputs its reverse  $\sigma_n \dots \sigma_1$ , for all  $\sigma_i \in \Sigma$ . The transduction  $\text{copy}$  takes  $u$  and returns  $uu$ .

Let  $\Sigma$  be an alphabet and  $\underline{\Sigma} = \{\underline{\sigma} \mid \sigma \in \Sigma\}$ . Given a word  $u = \sigma_1 \dots \sigma_n$  and a position  $i \in \text{Pos}(u)$ , we let  $\text{under}_i(u) = \sigma_1 \dots \sigma_{i-1} \underline{\sigma_i} \sigma_{i+1} \dots \sigma_n$ . The transduction  $\text{square}: \Sigma^* \rightarrow (\Sigma \cup \underline{\Sigma})^*$  is defined as  $\text{square}(u) = \text{under}_1(u) \dots \text{under}_{|u|}(u)$ . For example  $\text{square}(abc) = \underline{a}bc\underline{a}\underline{b}c\underline{a}\underline{b}\underline{c}$ .

► **Example 4** (Subwords). Let  $\text{sub}: \Sigma^* \rightarrow \Sigma^*$  be the transduction which enumerates all the subwords of a word in lexicographic order (with rightmost significant bit). For example  $\text{sub}(abc) = a.b.ab.c.ac.bc.abc$ . Note that  $\text{sub}$  has exponential growth.

► **Example 5** (Map). Let  $\Sigma$  be some alphabet and  $| \notin \Sigma$  be some separator symbol. Let  $\Sigma_{|} = \Sigma \cup \{| \}$ . Let  $f: \Sigma^* \rightarrow \Gamma^*$ . The transduction  $\text{map } f: \Sigma_{|}^* \rightarrow \Sigma_{|}^*$  takes any input word of the form  $u = u_1|u_2| \dots |u_n$  where  $u_i \in \Sigma^*$  for all  $i \in \{1, \dots, n\}$ , and returns  $f(u_1)|f(u_2)| \dots |f(u_n)$  (if all the  $f(u_i)$  are defined, otherwise  $(\text{map } f)(u)$  is undefined).

**Sequential and rational transductions.** Sequential transductions are transductions recognized by sequential transducers. A *sequential transducer* over some alphabets  $\Sigma$  and  $\Gamma$  (not necessarily disjoint) is a pair  $T = (A, \mu)$  where  $A = (Q, q_0, F, \delta)$  is a DFA over  $\Sigma$  and  $\mu: \text{dom}(\delta) \rightarrow \Gamma^*$  is a total function. We write  $q \xrightarrow{u/v}_T q'$  whenever there exists a sequence of states  $q_1 = q, q_2, \dots, q_{n+1} = q'$  such that  $q_1 \xrightarrow{u[1]}_A q_2 \dots q_n \xrightarrow{u[n]}_A q_{n+1}$  where  $n = |u|$ , and  $v = \mu(q_1, u[1]) \dots \mu(q_n, u[n])$ . The transduction  $f_T$  recognized by  $T$  is defined for all  $u \in L(A)$  by  $f_T(u) = v$  such that  $q_0 \xrightarrow{u/v}_T q_f \in F$ . Note that  $\text{dom}(f_T) = L(A)$ . We denote by **Seq** the class of sequential transductions. Like sequential transducers, a (non-deterministic, functional) *finite state transducer* is defined as a pair  $T = (A, \mu)$  but  $A$  can be non-deterministic, with the *functional* restriction: for all words  $u \in L(A)$ , the outputs of all the accepting runs over  $u$  are all equal. With this restriction,  $T$  recognizes a transduction  $f_T$ . A *rational transduction* is a transduction  $f_T$  for some  $T$ , and we denote by **Rat** the class of rational transductions [21].

**Regular and polyregular transductions.** The class of regular (resp. polyregular) transductions is the smallest class of transductions which is closed under composition of transductions and **map**, and contains the sequential transductions, **copy** and **rev** (resp. the sequential transductions, **rev** and **square**) [5, 9]. We denote by **PolyReg** the class of polyregular transductions.

### 3 MSO set interpretations, properties and limitations

#### MSO set interpretations

**Signatures, formulas and structures.** A *relational signature* (or simply signature) is a set  $\mathcal{S}$  of symbols together with a function  $\text{arity} : \mathcal{S} \rightarrow \mathbb{N}$ . We consider a set of *first-order variables* denoted by lower case letter  $x, y, z, \dots$  as well as a set of *second-order variables* denoted by upper case letter  $X, Y, Z, \dots$ . The MSO-formulas over signature  $\mathcal{S}$ , denoted by  $\text{MSO}[\mathcal{S}]$ , are denoted by the grammar  $\phi ::= \exists x \phi \mid \exists X \phi \mid \phi \wedge \phi \mid \neg \phi \mid X(x) \mid R(x_1, \dots, x_r)$ , where  $x, x_1, \dots, x_r$  are first-order variables,  $X$  is a second-order variable and  $R \in \mathcal{S}$  with  $\text{arity}(R) = r$ . We denote by  $\text{FO}[\mathcal{S}]$  the formulas which do not use second-order variables.

A *relational structure*  $u$  over signature  $\mathcal{S}$  is a set  $U$  called the *universe* of the structure, together with, for each symbol  $R \in \mathcal{S}$  of arity  $r$ , an *interpretation*  $R^u \subseteq U^r$ .

**Regularity preserving.** A function from  $\mathcal{S}$ -structures to  $\mathcal{T}$ -structures is called *regularity preserving* if the inverse image of an  $\text{MSO}[\mathcal{T}]$  definable set is  $\text{MSO}[\mathcal{S}]$  definable. We say that a class of functions is regularity preserving if all functions in the class are.

**Word structures.** The *word signature* over  $\Sigma$  is the tuple  $S_\Sigma = ((\sigma(x))_{\sigma \in \Sigma}, \leq(x, y))$  where  $\sigma(x)$  are unary predicate symbols and  $\leq(x, y)$ , usually written  $x \leq y$ , is a binary predicate symbol. Any word  $u$  can be naturally associated with an  $S_\Sigma$ -structure  $\tilde{u} = (U, (\sigma^{\tilde{u}})_{\sigma \in \Sigma}, \leq^{\tilde{u}})$  where  $U = \text{Pos}(u)$ ,  $\sigma^{\tilde{u}}$  is a set of positions labeled  $\sigma$ , for all  $\sigma \in \Sigma$ , and  $\leq^{\tilde{u}}$  is the natural (linear) order on  $\text{Pos}(u)$ . We write  $u$  instead of  $\tilde{u}$  if it is clear from the context that  $u$  is an  $S_\Sigma$ -structure. A *word structure* over  $\Sigma$  is an  $S_\Sigma$ -structure isomorphic to some  $\tilde{u}$ . Note that being a word structure is FO definable.

► **Definition 6** (MSO set interpretations [11]). An *MSO set interpretation (MSOSI)*  $T$  from  $\mathcal{S}$ -structures to  $\mathcal{T}$ -structures, is given by  $k \in \mathbb{N} \setminus \{0\}$  called the *dimension*, a domain formula  $\phi_{\text{dom}} \in \text{MSO}[\mathcal{S}]$ , an output universe formula  $\phi_{\text{univ}}(\bar{X}) \in \text{MSO}[\mathcal{S}]$ , and for each symbol  $R \in \mathcal{T}$  of arity  $r$  a formula  $\phi_R(\bar{X}_1, \dots, \bar{X}_r) \in \text{MSO}[\mathcal{S}]$ , where  $\bar{X}, \bar{X}_1, \dots$  are  $k$ -tuples of variables.

The semantics of  $T$  is a partial transduction  $f_T$  from  $\mathcal{S}$ -structures to  $\mathcal{T}$ -structures. The domain of  $f_T$  is the set of structures  $u$  such that  $u \models \phi_{\text{dom}}$ . Given such a  $u$  with universe  $U$ , we define its image  $v = f_T(u)$  as the structure with universe  $V = \{\bar{P} \in (2^U)^k \mid u \models \phi_{\text{univ}}(\bar{P})\}$ , and for each  $R \in \mathcal{T}$  of arity  $r$ , the interpretation  $R^v = \{(\bar{P}_1, \dots, \bar{P}_r) \in V^r \mid u \models \phi_R(\bar{P}_1, \dots, \bar{P}_r)\}$ . We say that an MSOSI is (finite) word-to-word if its domain and co-domain only contain word structures over some respective alphabets  $\Sigma, \Gamma$ .

► **Remark 7.** Given an MSOSI from  $S_\Sigma$ -structures to  $S_\Gamma$ -structures, one can restrict the domain formula to word structures whose image are word structures. This is because being a linear order is FO-definable.

► **Example 8.** The transduction  $\text{sub}$  of Example 4 is definable by the MSOSI  $T = (k = 2, \phi_{\text{dom}} = \top, \phi_{\text{univ}}(X, Y), (\phi_\sigma(X, Y))_{\sigma \in \Sigma}, \phi_\leq(X, Y))$ . The main idea is to let  $X$  range over all possible subsets, and  $Y$  range over all possible singletons  $\{y\}$  such that  $y \in X$ . Let  $\text{sing}(Y, y) = Y(y) \wedge \forall y'(Y(y') \rightarrow y' = y)$ . It holds true iff  $Y = \{y\}$ . Then,  $\phi_{\text{univ}}(X, Y) = \exists y(\text{sing}(Y, y) \wedge X(y))$ . The label of an output position  $(X, \{y\})$  is the label of the input position  $y$ , i.e.  $\phi_\sigma(X, Y) = \exists y(\text{sing}(Y, y) \wedge \sigma(y))$ . Finally, any two output positions  $(X_1, \{y_1\})$ ,  $(X_2, \{y_2\})$  are ordered lexicographically (with rightmost significant bit): if  $X_1 = X_2$ , then  $y_1 \leq y_2$ , otherwise, the smallest mismatching position  $x$  (i.e. a position in the symmetric difference of  $X_1$  and  $X_2$ ) must be in  $X_1$ . Those properties are easily expressible in MSO.



**MSO transductions, MSO and FO interpretations.** An MSO interpretation (MSOI) is an MSOSI whose free set variables are restricted to be singleton sets. This can be syntactically enforced in the universe formula  $\phi_{\text{univ}}$ , as being a singleton is an MSO definable property. Equivalently, MSOI are defined as MSOSI but instead the free variables are first-order. Note that transductions realized by MSOI have only polynomial growth. An FO interpretation (FOI) is an MSOI whose formulas are all FO-formulas. Finally an MSO transduction (MSOT) is (roughly<sup>7</sup> speaking) an MSO interpretation of dimension 1. MSOT capture exactly the class of regular transductions [1, 13, 16, 23].

The following theorem is at the core of the theory of set interpretations, and automatic structures. It holds in all generality, and furthermore the compositions can be done by simple formula substitutions.

► **Theorem 9** ([11, Proposition 2.4]). *MSO set interpretations are effectively closed under pre-composition by MSOT and post-composition by FOI.*

**Exponential versus polynomial growth.** There is a dichotomy for the growth of set interpretations over words, deeply connected to the similar dichotomy for the automata ambiguity, between *exponential* and *polynomial growths*. Moreover for polynomial growth transductions, the level of growth exactly coincides with the minimum dimension of an MSOSI defining the transduction. The result holds in the more general case of trees.

► **Theorem 10** ([20, Theorem 1.5], [6, Theorem 2.3]). *A set interpretation over words has growth either  $2^{\Theta(n)}$ , or  $\Theta(n^k)$  for some  $k \in \mathbb{N}$ , and this can be computed in PTIME. In the latter case<sup>8</sup>, one can compute an equivalent MSOI of dimension  $k$ .*

Quite a lot is known about word-to-word set interpretations with polynomial growth, which are called polyregular transductions and enjoy many different characterizations [8, Theorem 7].

**Regularity preserving.** An open question on word-to-word MSOSI is whether they are regularity preserving. This can actually be formulated in terms of closure properties.

► **Proposition 11.** *The following are equivalent:*

- *Word-to-word MSOSI are regularity preserving,*
- *The class of word-to-word MSOSI is closed under post-composition with transductions computed by Mealy machines (see [24] for a definition of Mealy machines),*
- *The class of word-to-word MSOSI is closed under post-composition with polyregular transductions.*

## Automatic transductions

We describe an automata-based presentation of MSOSI, which we call *automatic transductions*. Algorithmically speaking, it is more amenable to efficient processing, as it is based on automata instead of MSO, and it makes the connection between automatic structures and set interpretations more obvious.

► **Definition 12.** *An automatic transduction (AT for short) from  $\Sigma^*$  to  $\mathcal{T}$ -structures is described as a tuple  $T = (\Sigma, B, A_{\text{dom}}, A_{\text{univ}}, (A_R)_{R \in \mathcal{T}})$  where:*

- *$B$  is a finite alphabet describing a work alphabet*

<sup>7</sup> Classically, one adds a bounded number of copies of the input to get the full class of MSOT.

<sup>8</sup> Note that to get this tight correspondence, we need to allow a bounded number of copies of the input, see [20, Definition 4.3].

- $A_{\text{dom}}$  is an automaton over  $\Sigma$  recognizing the domain of the transduction,
- $A_{\text{univ}}$  is an automaton over  $\Sigma \times B$ . Words accepted by  $A_{\text{univ}}$  are called configurations,
- for each  $R \in \mathcal{T}$  of arity  $r$ ,  $A_R$  is an automaton over  $\Sigma \times B^r$  describing tuples of the relation  $R$ .

Given a word  $u \in \Sigma^*$ , the output  $\mathcal{T}$ -structure  $v = f_T(u)$  is defined, whenever  $u \in L(A_{\text{dom}})$ , as follows: its universe is the set  $V = \{x \in B^* \mid u \otimes x \in L(A_{\text{univ}})\}$ ; a predicate symbol  $R \in \mathcal{T}$  of arity  $r$  is interpreted as  $R^v = \{(x_1, \dots, x_r) \in V^r \mid u \otimes x_1 \otimes \dots \otimes x_r \in L(A_R)\}$ .

► **Remark 13.** Automatic transductions are essentially identical to MSOSI, except restricted to input word structures, where one can leverage the classical equivalence between MSO and automata. They can be naturally generalized to work over input structures such as  $\omega$ -words, trees and infinite trees, giving rise to the notions of  $\omega$ -automatic, tree-automatic and  $\omega$ -tree-automatic transductions. Note that an  $\omega$ -automatic structure is precisely an  $\omega$ -automatic transduction whose domain is a single infinite word  $a^\omega$ .

**Transduction/structure correspondence.** In [18], we show a correspondence between word-to-word automatic transductions and automatic  $\omega$ -words (defined therein), thanks to which we obtain the following proposition.

► **Proposition 14.** *If word-to-word MSOSI are effectively regularity preserving, then automatic  $\omega$ -words have a decidable MSO theory<sup>9</sup>.*

This entails that a positive answer to Conjecture 2 would provide a positive answer to Conjecture 1, which has been open since [2].

## 4 Lexicographic transductions

As explained in the latter section, we do not know whether MSOSI are regularity preserving, and as a consequence of Proposition 14, proving that it enjoys this property would prove a long-standing conjecture of the theory of automatic structures [2]. In this section, we introduce a subclass of MSOSI which enjoys this property, called *lexicographic transductions*.

### Definition of lexicographic transductions

We first define this class in terms of closure of basic transductions, called *simple transductions*, under a *lexicographic map* operation. The connection with MSOSI is done at the end of this section (Subsection 4), via a corresponding subclass of automatic transductions.

**Simple transductions.** A *regular constant* (partial) transduction of type  $\Sigma^* \rightarrow \Gamma^*$  is an expression of the form  $L \triangleright w$ , where  $L$  is a regular language over  $\Sigma$  and  $w$  is a word in  $\Gamma^*$ , such that for all  $u \in \Sigma^*$ ,  $(L \triangleright w)(u)$  is defined only if  $u \in L$ , by  $(L \triangleright w)(u) = w$ . A *simple transduction*<sup>10</sup>  $f$  is a finite union of regular constant transductions whose codomain only contains words of length at most 1. A simple transduction  $f: \Sigma^* \rightarrow \Gamma^*$  is denoted by  $f = \sum_{i=1}^n L_i \triangleright w_i$  such that  $L_1, \dots, L_n \subseteq \Sigma^*$  are pairwise disjoint regular languages, and  $w_1, \dots, w_n \in \Gamma^{\leq 1}$ .

<sup>9</sup> One could actually prove the stronger implication that any automatic  $\omega$ -word has a canonical presentation, as in [2].

<sup>10</sup> It is a restriction of the known class of *recognizable transduction* to output words of length at most 1.



**Lexicographic enumerators.** An ordered alphabet is a pair  $\lambda = (B, \prec)$  such that  $B$  is finite set and  $\prec$  is a linear order over  $B$ . The order  $\prec$  is extended lexicographically (using the same notation) to words of same length over  $B$ , with most significant letter to the right: for all  $n$  and all  $u, v \in B^n$ ,  $u \prec v$  if there exists a position  $i \leq n$  such that  $u[i] \prec v[i]$  and for all  $i < j \leq n$ ,  $u[j] = v[j]$ . Note that  $\prec$  is a total order over  $B^n$ , for all  $n$ . We denote  $\text{succ}_\lambda : B^* \rightarrow B^*$  the successor function on  $B^*$  induced by  $\prec$ .

Remind that  $|$  is a fixed separator symbol. The  $\lambda$ -lexicographic enumerator is the function

$$\begin{aligned} \text{lex-enum}_\lambda : \bigcup_{\Sigma, \Gamma \text{ alphabets}} \Sigma^* &\rightarrow ((\Sigma \times B)^* |)^* (\Sigma \times B)^* \\ w &\mapsto (w \otimes u_1) | (w \otimes u_2) | \dots | (w \otimes u_k) \end{aligned}$$

where  $|w| = |u_1| = \dots = |u_k|$ ,  $u_1$  is minimal for  $\prec$ ,  $u_k$  is maximal for  $\prec$  and for all  $1 \leq i < k$ ,  $u_{i+1} = \text{succ}_\lambda(u_i)$ . Note that  $k = |B|^{|w|}$ .

► **Example 15.** Let  $\Sigma = \{a, b\}$  let  $\lambda = (B, \prec)$  be a finite order with  $B = \{0, 1\}$  and  $0 \prec 1$ . For all  $\sigma \in \Sigma$  and  $b \in B$ , we write  $\begin{smallmatrix} \sigma \\ b \end{smallmatrix}$  instead of  $(\sigma, b)$  and  $\begin{smallmatrix} \Sigma \\ b \end{smallmatrix}$  to denote the set of pairs  $(\sigma, b)$  for all  $\sigma \in \Sigma$ . Then  $\text{lex-enum}_\lambda(abb) = \begin{smallmatrix} a & b & b \\ 0 & 0 & 0 \end{smallmatrix} | \begin{smallmatrix} a & b & b \\ 1 & 0 & 0 \end{smallmatrix} | \begin{smallmatrix} a & b & b \\ 0 & 1 & 0 \end{smallmatrix} | \begin{smallmatrix} a & b & b \\ 1 & 1 & 0 \end{smallmatrix} | \begin{smallmatrix} a & b & b \\ 0 & 0 & 1 \end{smallmatrix} | \begin{smallmatrix} a & b & b \\ 1 & 0 & 1 \end{smallmatrix} | \begin{smallmatrix} a & b & b \\ 0 & 1 & 1 \end{smallmatrix} | \begin{smallmatrix} a & b & b \\ 1 & 1 & 1 \end{smallmatrix}$ .

**MapLex combinator.** Let  $\lambda = (B, \prec)$  be an ordered alphabet. We define the function

$$\text{maplex}_\lambda : \bigcup_{\Sigma, \Gamma \text{ alphabets}} ((\Sigma \times B)^* \rightarrow \Gamma^*) \rightarrow \Sigma^* \rightarrow \Gamma^*$$

such that for all  $\Sigma, \Gamma$  alphabets, all  $f : (\Sigma \times B)^* \rightarrow \Gamma^*$  and  $u \in \Sigma^*$ ,

$$\text{maplex}_\lambda f u = f(v_1)f(v_2)\dots f(v_k)$$

where  $\text{lex-enum}_\lambda(u) = v_1|v_2|\dots|v_k$ . Note that  $u$  is in the domain of  $\text{maplex}_\lambda f$  if and only if  $v_1, \dots, v_k$  are all in the domain of  $f$ . We write  $\text{maplex}$  when  $\lambda$  is clear from the context.

► **Definition 16** (Lexicographic transductions). *Lexicographic transductions, denoted by  $\text{Lex}$ , are defined inductively by  $\text{Lex}_0$  the class of simple transductions and  $\text{Lex}_{k+1} = \{\text{maplex}_\lambda f \mid f \in \text{Lex}_k, \lambda \text{ ordered alphabet}\}$ . Elements of  $\text{Lex}_k$  are called  $k$ -lexicographic transductions.*

► **Lemma 17.** *For all  $f \in \text{Lex}$ , its domain  $\text{dom}(f)$  is regular.*

**Proof.** Any  $\text{Lex}$  transduction  $f : \Sigma^* \rightarrow \Gamma^*$  is equal to  $\text{maplex}_{\lambda_1} (\text{maplex}_{\lambda_2} \dots (\text{maplex}_{\lambda_k} s) \dots)$  for some  $k \geq 0$ , some ordered alphabets  $(\lambda_i = (B_i, \prec_i))_i$  and some simple transduction  $s : (\Sigma \times B_1 \times \dots \times B_k)^* \rightarrow \Gamma^*$ . Then,  $f$  is defined on  $u \in \Sigma^*$  iff for all  $1 \leq i \leq k$  and all  $b_i \in B_i^{|u|}$ ,  $s(u \otimes b_1 \otimes \dots \otimes b_k)$  is defined. Now, observe that  $\text{dom}(f)$  is the complement of the  $\Sigma$ -projection of the complement of  $\text{dom}(s)$ . This entails the result as  $\text{dom}(s)$  is regular and regular languages are closed under morphisms (and complement). ◀

► **Remark 18.** A simple transduction  $s \in \text{Lex}_0$ , has growth  $O(1)$ , and  $\text{maplex}_\lambda s$  has growth  $O(|B|^n)$ , for  $\lambda = (B, \lambda)$ . One application of  $\text{maplex}$  can thus cause an exponential blowup. Note however that these exponentials don't compose, but multiply with extra applications of  $\text{maplex}$ . Using the same notations as in the above proof,  $f$  has growth  $O(|B_1 \times \dots \times B_k|^n)$ .

► **Example 19** (Identity and Reverse). Take  $\lambda = (B, \prec)$  and  $\lambda' = (B, \prec')$  with  $B = \{0, 1\}$ ,  $0 \prec 1$ , and  $1 \prec' 0$ . For all  $\sigma \in \Sigma$ , let  $L_\sigma = (\begin{smallmatrix} \Sigma \\ 0 \end{smallmatrix})^* (\begin{smallmatrix} \sigma \\ 1 \end{smallmatrix}) (\begin{smallmatrix} \Sigma \\ 0 \end{smallmatrix})^*$  and  $L_\epsilon = (\Sigma \times B)^* \setminus (\bigcup_\sigma L_\sigma)$ .

$$\text{id} = \text{maplex}_\lambda (L_\epsilon \triangleright \epsilon + \sum_{\sigma \in \Sigma} L_\sigma \triangleright \sigma) \quad \text{rev} = \text{maplex}_{\lambda'} (L_\epsilon \triangleright \epsilon + \sum_{\sigma \in \Sigma} L_\sigma \triangleright \sigma)$$

This is illustrated below on input  $abc$ , with the output of the simple function below every word of the enumeration.

$$\begin{array}{cccccccc}
 \text{id} & \underbrace{\begin{smallmatrix} a & b & c \\ 0 & 0 & 0 \end{smallmatrix}}_{\epsilon} & \underbrace{\begin{smallmatrix} a & b & c \\ 1 & 0 & 0 \end{smallmatrix}}_a & \underbrace{\begin{smallmatrix} a & b & c \\ 0 & 1 & 0 \end{smallmatrix}}_b & \underbrace{\begin{smallmatrix} a & b & c \\ 1 & 1 & 0 \end{smallmatrix}}_{\epsilon} & \underbrace{\begin{smallmatrix} a & b & c \\ 0 & 0 & 1 \end{smallmatrix}}_c & \underbrace{\begin{smallmatrix} a & b & c \\ 1 & 0 & 1 \end{smallmatrix}}_{\epsilon} & \underbrace{\begin{smallmatrix} a & b & c \\ 0 & 1 & 1 \end{smallmatrix}}_{\epsilon} & \underbrace{\begin{smallmatrix} a & b & c \\ 1 & 1 & 1 \end{smallmatrix}}_{\epsilon} \\
 \text{rev} & \underbrace{\begin{smallmatrix} a & b & c \\ 1 & 1 & 1 \end{smallmatrix}}_{\epsilon} & \underbrace{\begin{smallmatrix} a & b & c \\ 0 & 1 & 1 \end{smallmatrix}}_{\epsilon} & \underbrace{\begin{smallmatrix} a & b & c \\ 1 & 0 & 1 \end{smallmatrix}}_{\epsilon} & \underbrace{\begin{smallmatrix} a & b & c \\ 0 & 0 & 1 \end{smallmatrix}}_c & \underbrace{\begin{smallmatrix} a & b & c \\ 1 & 1 & 0 \end{smallmatrix}}_{\epsilon} & \underbrace{\begin{smallmatrix} a & b & c \\ 0 & 1 & 0 \end{smallmatrix}}_b & \underbrace{\begin{smallmatrix} a & b & c \\ 1 & 0 & 0 \end{smallmatrix}}_a & \underbrace{\begin{smallmatrix} a & b & c \\ 0 & 0 & 0 \end{smallmatrix}}_{\epsilon}
 \end{array}$$

► **Example 20 (Morphisms).** Let  $\phi_a : u \in \{a, b\}^* \rightarrow a^*$  be the morphism defined by  $\phi_a(a) = a$  and  $\phi_a(b) = \epsilon$ . We have  $\phi_a \in \text{Lex}_1$ . It suffices to take  $B = \{0, 1\}$  with  $0 \prec 1$ . Then, let  $L_a = \left(\frac{\Sigma}{0}\right)^* \left(\frac{a}{1}\right) \left(\frac{\Sigma}{0}\right)^*$ , and  $L_\epsilon = \overline{L_a}$ . Then  $\phi_a = \text{maplex}(L_a \triangleright a + L_\epsilon \triangleright \epsilon)$ . More generally, if  $\psi : \Sigma^* \rightarrow \Gamma^*$  is an arbitrary morphism, we show that  $\psi \in \text{Lex}_1$ . Note that  $\psi$  may transform a single letter into several letters, while simple transductions output at most one letter. To overcome this difference, we consider a larger linearly ordered set. Let  $M = \max_{\sigma \in \Sigma} |\psi(\sigma)|$ . If  $M = 0$ , then  $\psi$  is the constant transduction which outputs  $\epsilon$ , so  $\psi \in \text{Lex}_0$ . Otherwise, let  $\lambda_M = (B_M, <)$  with  $B_M = \{0, 1, \dots, M\}$  naturally ordered. Let  $I : \Gamma \rightarrow 2^{\Sigma \times \mathbb{N}}$  such that for all  $\gamma \in \Gamma$ ,  $I(\gamma)$  is the set of pairs  $(\sigma, i)$  such that  $\psi(\sigma)[i] = \gamma$ . Note that for all  $\gamma \in \Gamma$ ,  $I(\gamma) \subseteq \Sigma \times \{1, \dots, M\}$ . Define  $L_\gamma$  as the set given by the regexp  $\bigcup_{(\sigma, i) \in I(\gamma)} \left(\frac{\Sigma}{0}\right)^* \left(\frac{\sigma}{i}\right) \left(\frac{\Sigma}{0}\right)^*$  and  $L_\epsilon$  the complement of the union of all  $L_\gamma$ . Then  $\psi = \text{maplex}_{\lambda_M}(L_\epsilon \triangleright \epsilon + \sum_{\gamma \in \Gamma} L_\gamma \triangleright \gamma)$ .

Example 20 can be generalized to sequential transductions.

► **Lemma 21.**  $\text{Seq} \subseteq \text{Lex}_1$ .

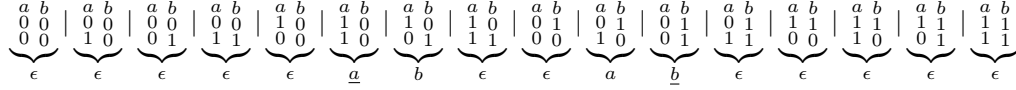
► **Example 22 (Domain restriction).** Let  $k \geq 0$ . Given  $f : \Sigma^* \rightarrow \Gamma^*$  a transduction in  $\text{Lex}_k$  and  $L \subseteq \Sigma^*$  a regular language, the transduction  $f|_L : u \mapsto f(u)$  if  $u \in \text{dom}(f) \cap L$  is in  $\text{Lex}_k$ . We show this inductively on  $k$ : it is clear for  $f \in \text{Lex}_0$ . Assume  $f = \text{maplex}_\lambda g$  with  $\lambda = (B, <)$  and let  $\pi_\Sigma : (\Sigma \times B)^* \rightarrow \Sigma^*$  be the natural projection morphism. Then  $f|_L = \text{maplex}_\lambda g|_{\pi_\Sigma^{-1}(L)}$ , which proves that  $\text{Lex}_k$  is closed under domain restriction.

► **Example 23 (Subwords).** We show that  $\text{sub} \in \text{Lex}_2$  (see Example 4 for the definition of  $\text{sub}$ ). We take  $\lambda = (B, <)$ , with  $B = \{0, 1\}$  and define the following morphism  $\text{del}_0 : (\Sigma \times B)^* \rightarrow \Sigma^*$  by  $\text{del}_0(\sigma, 0) = \epsilon$  and  $\text{del}_0(\sigma, 1) = \sigma$ . We can then show  $\text{sub} = \text{maplex}_\lambda \text{del}_0$ . From Ex. 20, morphisms are in  $\text{Lex}_1$ , so  $\text{sub} \in \text{Lex}_2$ .

► **Example 24 (Square, illustrated on Fig. 2).** The transductions  $\text{square}$  and  $\text{under}_i$  have been defined in Ex. 3. We show that  $\text{square} \in \text{Lex}_2$ . Let  $\lambda = (B, <)$  with  $B = \{0, 1\}$  and let  $f : (\Sigma \times B)^* \rightarrow (\Sigma \cup \underline{\Sigma})^*$  such that for all  $u \in \Sigma^*$  of length  $n$ , for all  $1 \leq i \leq n$ ,  $f(u \otimes (0^{i-1}10^{n-i})) = \text{under}_i(u)$ , and for  $b \notin 0^*10^*$ ,  $f(u \otimes b) = \epsilon$ . It holds that  $\text{square} = \text{maplex}_\lambda f$ , because  $0^{i-1}10^{n-i} < 0^{j-1}10^{n-j}$  for all  $i < j$ . It remains to show that  $f \in \text{Lex}_1$ . It is because  $f = \text{maplex}_\lambda g$  for  $g : (\Sigma \times B^2)^* \rightarrow \Sigma^*$  the following simple transduction: for all  $u \otimes b_1 \otimes b_2 \in (\Sigma \times B^2)^*$ , if  $b_1 \notin 0^*10^*$  or  $b_2 \notin 0^*10^*$ ,  $g(u \otimes b_1 \otimes b_2) = \epsilon$ , otherwise let  $i$  be the unique position at which 1 occurs in  $b_1$  and  $j$  the unique position at which a 1 occurs in  $b_2$ . If  $i = j$ , then  $g(u \otimes b_1 \otimes b_2) = \underline{u[j]}$ , otherwise  $g(u \otimes b_1 \otimes b_2) = u[j]$ . Since those properties are regular,  $g$  is a simple transduction.

## Presentation as automatic transductions

We give an alternative presentation in terms of automatic transductions. Let  $k \geq 1$  be a positive integer, and  $\bar{\lambda} = ((B_1, \prec_1), \dots, (B_k, \prec_k))$  be a  $k$ -tuple of ordered alphabets and let  $B = B_1 \times \dots \times B_k$ . We define the associated  $k$ -lexicographic order for words of the same length over  $B^*$  by  $u \prec_{\bar{\lambda}} v$  if  $u = u_1 \otimes \dots \otimes u_k$ ,  $v = v_1 \otimes \dots \otimes v_k$ , and there is  $i \in \{1, \dots, k\}$  such that  $u_i \prec_i v_i$  and for all  $j < i$ ,  $u_j = v_j$ .



■ **Figure 2** Equality square =  $(\text{maplex}_\lambda \text{ maplex}_\lambda g)$  illustrated on input  $ab$ , with the results of applying  $g$  underneath.

► **Definition 25.** Let  $\bar{\lambda} = ((B_1, \prec_1), \dots, (B_k, \prec_k))$  be a  $k$ -tuple of ordered alphabets, let  $B = B_1 \times \dots \times B_k$  and let  $\prec_{\bar{\lambda}}$  be the associated  $k$ -lexicographic order.

A  $k$ -lexicographic automatic transducer over the alphabet  $B$  is an automatic transducer with work alphabet  $B$  such that the order is exactly  $\prec_{\bar{\lambda}}$ . A transduction is said to be  $k$ -lexicographic automatic if it can be defined by a  $k$ -lexicographic automatic transducer. We denote by  $\text{AT}_{k\text{-Lex}}$  the class of  $k$ -lexicographic automatic transductions and by  $\text{AT}_{\text{Lex}}$  the union of these, which we call *lexicographic automatic transductions*.

The next proposition is rather immediate.

► **Proposition 26** ( $\text{AT}_{k\text{-Lex}} = \text{Lex}_k$ ). For all  $k \geq 1$ , a transduction is  $k$ -lexicographic iff it is  $k$ -lexicographic automatic.

In [18], we make a connection between  $k$ -lexicographic automatic transductions and  $k$ -lexicographic automatic  $\omega$ -words [2]. In [2, Theorem 6.1], the author shows that  $k$ -lexicographic automatic  $\omega$ -words form a strict hierarchy and provides explicit witnesses for each level of the hierarchy. As a consequence of Proposition 26, we obtain that  $k$ -lexicographic transductions form a strict hierarchy, as stated by the following proposition.

► **Proposition 27.** For all  $k \geq 1$ ,  $\text{Lex}_k \subsetneq \text{Lex}_{k+1}$ .

## 5 Nested marble transducers

We introduce in this section a transducer model, called *nested marble transducers*, and show that the class of transductions it recognizes is exactly the class of lexicographic transductions. Nested marble transducers generalize marble transducers [12, 15]. A marble transducer belongs to the family of transducers with an unbounded number of pebbles (of finitely many colours), with the following restriction: whenever it moves left, it has to drop a pebble, and whenever it moves right, it has to lift a pebble. The term *marble* is meant to emphasize this restriction. A nested marble transducer of level  $k \geq 1$  behaves like a marble transducer which can call, when reading the leftmarker  $\vdash$ , a nested marble transducer of level  $k-1$ . A nested marble transducer of level 0 is what we call a simple transducer. It is just a DFA with an output function on its accepting states, so it realizes a transduction whose range is finite.

► **Definition 28** (Simple transducers). Let  $\Sigma, \Gamma$  be finite sets (not necessarily disjoint). A  $(\Sigma, \Gamma)$ -simple transducer is a pair  $T = (A, \mu)$  where  $A = (Q, q_0, F, \delta)$  is a DFA over  $\Sigma \cup \{\vdash, \dashv\}$  and  $\mu : F \rightarrow \Gamma^{\leq 1}$  is a total function.

We define two semantics for  $T$ , an operational semantics  $f_T^{\text{op}} : Q \times \Sigma^* \rightarrow \Gamma^* \times F$  which takes as input a word and also a state from which the computation starts, and returns a word and the state reached when the computation ends, if it is accepting. Otherwise  $f_T^{\text{op}}$  is not defined. Formally,  $f_T^{\text{op}}(q, u)$  is defined for all  $u$  such that  $q \xrightarrow{\vdash u \dashv}_A q_f$  for some  $q_f \in F$ , by  $f_T^{\text{op}}(q, u) = (\mu(q_f), q_f)$ .

From the operational semantics, we also define the transduction  $f_T : \Sigma^* \rightarrow \Gamma^*$  recognized by  $T$  by applying the operational semantics from the initial state, and projecting away the final state, i.e.  $f_T(u) = \pi_1(f_T^{\text{op}}(q_0, u))$ , where  $\pi_1$  is the projection on the first component.

► **Definition 29** (Nested marble transducers from  $\Sigma$  to  $\Gamma$ ). A  $(0, \Sigma, \Gamma)$ -nested marble transducer is a  $(\Sigma, \Gamma)$ -simple transducer. For  $k \geq 1$ , a  $(k, \Sigma, \Gamma)$ -nested marble transducer is a tuple  $T = (\Sigma, \Gamma, C, c_0, Q_T, q_0, F_T, \delta, \delta_{\text{call}}, \delta_{\text{ret}}, \mu, T')$  where:

- $C$  is a finite set of (marble) colors,  $c_0$  is an initial color;
- $Q_T$  is a finite set of states,  $q_0$  is an initial state, and  $F_T$  a set of accepting states;
- $T'$  is a  $(k-1, \Sigma \times C, \Gamma)$ -nested marble transducer with set of states  $Q_{T'}$  and set of accepting states  $F_{T'}$ ;
- $\delta : Q_T \times (\Sigma \cup \{\neg\}) \times C \rightarrow (C \cup \{\perp\}) \times Q_T$  is a transition function;
- $\delta_{\text{call}} : Q_T \times C \rightarrow Q_{T'}$  is a call function;  $\delta_{\text{ret}} : Q_T \times C \times F_{T'} \rightarrow Q_T$  is a return function;
- $\mu : \text{dom}(\delta) \rightarrow \Gamma^*$  is an output function.

We use  $(k, \Sigma, \Gamma)$ -NMT (or just  $k$ -NMT if  $\Sigma, \Gamma$  are clear from the context) as a shortcut for  $(k, \Sigma, \Gamma)$ -nested marble transducer.  $T'$  is the assistant NMT and  $k$  the level of  $T$ . Finally, we often say marble instead of marble colour.

We now define the semantics informally. The reading head of  $T$  is initially placed on the rightmost position labeled  $\neg$ , marked with a marble of color  $c_0$ , in state  $q_0$ . Transitions work as follows: suppose the current state is  $q$  and the reading head is on some position  $i$  labeled by  $\sigma \in \Sigma \cup \{\neg, \vdash\}$  and by some marble of color  $c \in C$ . Whatever transition in  $\delta$  can be applied, some output word is produced by  $T$  according to  $\mu$ . Then there are three cases:

1. if  $\sigma \in \Sigma \cup \{\neg\}$  and  $\delta(q, \sigma, c) = (c', q')$  where  $c' \in C$ , then the reading head moves to position  $(i-1)$  in state  $q'$  and a marble of color  $c'$  is placed (on position  $i-1$ );
2. if  $\sigma \in \Sigma$  and  $\delta(q, \sigma, c) = (\perp, q')$ , then  $T$  lifts the current marble and moves its reading head to position  $i+1$  in state  $q'$ ;
3. if  $\sigma = \vdash$  then  $T$  calls  $T'$  initialized with state  $\delta_{\text{call}}(q, c)$ , on the input word annotated with marbles. When  $T'$  finishes its computation in some accepting state  $q'$ ,  $T$  lifts marble  $c$ , moves its reading head to position 1 and continues its computation from state  $\delta_{\text{ret}}(q, c, q')$ .

The (operational) semantics of  $T$  is a function  $f_T^{\text{op}} : Q_T \times \Sigma^* \rightarrow \Gamma^* \times F_T$ , that we define inductively. The case  $k = 0$  has been defined after Definition 28. If  $k \geq 1$  and  $T = (\Sigma, \Gamma, C, c_0, Q_T, q_0, F_T, \delta, \delta_{\text{call}}, \delta_{\text{ret}}, \mu, T')$  then we assume  $f_{T'}^{\text{op}} : Q_{T'} \times (\Sigma \times C)^* \rightarrow \Gamma^* \times F_{T'}$  to be defined inductively. Let us now define  $f_T^{\text{op}}$ . A *configuration* of  $T$  over a word  $u \in \Sigma^*$  is a triple  $(q, i, v)$  such that  $q$  is the current state,  $i \in \text{Pos}(u) \cup \{0, n+1\}$  is the current position (where  $n = |u|$ ), and  $v \in C^*$  is an annotation of the suffix  $(\vdash u \neg)[i:n+1]$ . We define a labeled successor relation  $(q, i, cv) \xrightarrow{w}_T (q', i', v')$ , between any two configurations where  $c \in C$ , labeled by  $w \in \Gamma^*$ , whenever one of the following cases hold:

1.  $1 \leq i \leq n+1$ ,  $\delta(q, c) = (c', q')$ ,  $i' = i-1$ ,  $v' = c'cv$  and  $w = \mu(q, c)$ ;
2.  $1 \leq i \leq n$ ,  $\delta(q, c) = (\perp, q')$ ,  $i' = i+1$ ,  $v' = v$  and  $w = \mu(q, c)$ ;
3.  $i = 0$ ,  $f_{T'}^{\text{op}}(\delta_{\text{call}}(q, c), (\vdash u \neg) \otimes cv) = (w, p)$ ,  $q' = \delta_{\text{ret}}(q, c, p)$ ,  $i' = 1$  and  $v' = v$ .

The function  $f_T^{\text{op}} : Q_T \times \Sigma^* \rightarrow \Gamma^* \times F_T$  recognized by  $T$  is defined, for all  $q \in Q_T$  and all  $u \in \Sigma^*$  such that there exists a sequence of configurations over  $u$ :  $\nu_0 = (q, n+1, c_0) \xrightarrow{w_1}_T \nu_1 \xrightarrow{w_2}_T \nu_3 \dots \nu_{k-1} \xrightarrow{w_k}_T \nu_k$  where the state  $q_f$  of  $\nu_k$  is accepting (i.e. in  $F$ ) and the states of configurations  $\nu_i$ ,  $i < k$ , are non-accepting, by  $f_T^{\text{op}}(q, u) = (w_1 \dots w_k, q_f)$ .

The transduction  $f_T : \Sigma^* \rightarrow \Gamma^*$  recognized by  $T$  is defined as the projection of  $f_T^{\text{op}}(q_0, u)$  on  $\Sigma$  and  $\Gamma$ , i.e. if  $f_T^{\text{op}}(q_0, u) = (v, q_f)$  then  $f_T(u) = v$ . We denote by NMT the class of transductions recognizable by some  $(k, \Sigma, \Gamma)$ -NMT. The local size of an NMT is the number of its transitions, states and marbles. Its size is its local size plus the size the lower level NMT it calls. We similarly define the number of (resp. local number of) states/marbles/transitions.

► **Example 30.** We describe a 2-NMT  $T_2$  computing the transduction  $\text{sub}$  of Example 23. We recall that  $\text{sub} = \text{maplex}_\lambda \text{del}_0$  where  $\lambda = (B = \{0, 1\}, <)$  and  $\text{del}_0$  is a morphism. The transducer  $T_2$  behaves as a marble transducer which computes the  $\lambda$ -lexicographic enumerator  $\text{lex-enum}_\lambda$ , and whenever a full annotation of its input has been computed, it calls a 1-NMT  $T_1$  which computes  $\text{del}_0$ . Let us explain how  $T_2$  computes the next annotation in lexicographic order (which corresponds to the binary addition with most significant bit to the right). When the reading head of  $T_2$  is on the left marker, all input positions are marked with some pebble in  $B = \{0, 1\}$ . Then,  $T_2$  scans its input to the right (lifting all the pebbles it sees) until it reads a 0, replaces it by 1 and moves again to the left marker, dropping pebble 1 all the way back to the left marker. Only three states are needed. Initially,  $T_2$  drops pebble 0 on all positions, from right to left. We explain now how  $T_1$  works: it scans its input from right to left, and, whenever it reads an input  $(\sigma, b)$  with  $b = 1$ , outputs  $\sigma$ . Whenever  $T_1$  reaches the leftmarker, it calls a simple transducer  $T_0$  which does nothing but outputting  $\epsilon$ .

The following result states that NMT are closed under post-composition with Seq. To prove it, we strongly rely on the ability to pass state information through mappings  $\delta_{\text{call}}$  and  $\delta_{\text{ret}}$  to adapt a classical product construction of automata.

► **Lemma 31** ( $\text{Seq} \circ \text{NMT} \subseteq \text{NMT}$ ). *For all  $k \geq 0$ , all  $(k, \Sigma, \Gamma)$ -NMT  $T$  and all sequential transducer  $S$  over  $\Gamma, \Lambda$ , one can construct, in polynomial time, a  $(\max(k, 1), \Sigma, \Lambda)$ -NMT  $T'$  such that  $f_{T'} = f_S \circ f_T$ .*

**State-passing free nested marble transducers.** In the definition of NMT, there are two explicit forms of information-passing: state information can be passed from level  $k$  to level  $k-1$  through the function  $\delta_{\text{call}}$ , and state information can be passed from level  $k-1$  to level  $k$  via the function  $\delta_{\text{ret}}$ . In addition, there is an implicit one through the domain of assistant transducers: indeed, the definition of the semantics requires that all calls to assistant transducers do accept, hence the assistant transducer can influence the master transducer by rejecting a word. In this subsection, we prove that information-passing can be removed while preserving the computational power of  $k$ -NMT, however at the cost of increasing the size by a tower of exponentials of height  $k$ . While state-passing was useful to prove the closure under post-composition with sequential transductions (Lemma 31), it will be more convenient to consider state-passing free nested marble transducers in the sequel, in particular to prove that NMT recognize lexicographic transductions (Subsection 5).

► **Definition 32.** A state-passing free  $(k, \Sigma, \Gamma)$ -nested marble transducer ( $(k, \Sigma, \Gamma)$ -NMT<sub>spf</sub> for short), is either a simple transducer if  $k = 0$ , or, if  $k > 0$ , a  $(k, \Sigma, \Gamma)$ -NMT  $T = (\Sigma, \Gamma, C, c_0, Q_T, q_0, F, \delta, \delta_{\text{call}}, \delta_{\text{ret}}, \mu, T')$  such that

1.  $T'$  is a  $(k-1, \Sigma \times C, \Gamma)$ -NMT<sub>spf</sub> with set of states  $Q_{T'}$  and initial state  $q'_0$
2.  $\delta_{\text{call}}(q, c) = q'_0$  for all  $q \in Q_T$  and  $c \in C$
3.  $\delta_{\text{ret}}(q, c, q') = q$  for all  $q \in Q_T, c \in C$  and  $q' \in Q_{T'}$
4. calls to the assistant transducer  $T'$  always accept.

Since the functions  $\delta_{\text{call}}$  and  $\delta_{\text{ret}}$  play no role, we often omit them in the tuple denoting  $T$ .

► **Theorem 33** (State-passing removal). *For all  $k$ -NMT  $T$ , there exists an equivalent  $k$ -NMT<sub>spf</sub>  $T'$  whose size is  $k\text{-EXP}$  in that of  $T$ . This non-elementary blow-up is unavoidable.*

Before proving this result, we show a property on domains of NMT. A nested marble automaton  $A$  of level  $k$  is a nested marble transducer  $T$  of level  $k$  whose output function  $\mu$  is the constant function that always returns  $\epsilon$ . The language of  $A$  is defined as  $L(A) = \text{dom}(f_T)$ .

► **Lemma 34.** *A language is recognizable by a nested marble automaton of level  $k$  and  $n$  states iff it is recognizable by a finite automaton of size in  $k\text{-EXP}(n)$ . This non-elementary blow-up is unavoidable.*

**Proof sketch.** It is clear that any regular language is the domain of some simple transduction. Conversely, let  $A$  be a nested marble automaton of level  $k$ . If  $k = 0$ , it is obvious. If  $k \geq 1$ , let  $A = (\Sigma, C, c_0, Q_A, q_0, F, \delta, \delta_{\text{call}}, \delta_{\text{ret}}, A')$  where  $A'$  has level  $k-1$  (in the tuple, we have omitted the output alphabet and function, since they play no role). By IH, for all pairs  $(q_c, q_r)$  of states of  $A'$ , the language of words accepted by  $A'$  by a run starting in  $q_c$  and ending in  $q_r$  is regular, and can be described by some finite automaton  $D_{q_c, q_r}$  of size in  $(k-1)\text{-EXP}(n)$ .

We turn  $A$  into a marble automaton  $B$  (of level 1) such that  $L(B) = L(A)$ . The marbles of  $B$  are enriched with information on the states of automata  $D_{q_c, q_r}$ , for all pairs  $(q_c, q_r)$ , ensuring that  $B$  knows the state of all the automata  $D_{q_c, q_r}$  after reading the current suffix.  $B$  exploits this information to simulate  $A$  and whenever  $A$  calls  $A'$  with some initial state  $q_c$ , instead,  $B$  knows, if it exists, the state  $q_r$  of  $A'$  such that the current marked input is accepted by  $A'_{q_c, q_r}$ . If such a state exists, then it is unique as  $A'$  is deterministic, and  $B$  can bypass calling  $A'$  and directly apply its return transition. Otherwise,  $B$  stops.

The result follows as marble automata are known to recognize regular languages. The conversion of a marble automaton into a finite automaton is exponential both in the number of states and number of marbles (see e.g. [12, 15], as well as Theorem 5.4 of [22] for a detailed construction), yielding a tower of  $k$ -exponential inductively.

It can be shown that this non-elementary blowup is not avoidable, because first-order sentences on word structures (with one successor) with quantifier rank  $r$ , can be converted in an exponentially bigger nested marble automaton, while it is known that such sentences can be converted into an equivalent finite automaton of unavoidable size a tower of exponential of height  $r$  [19]. ◀

► **Corollary 35.** *Transductions recognized by nested marble transducers have regular domains.*

We are now ready to sketch the proof of Theorem 33.

**Proof sketch of Theorem 33.** There are two kinds of state-passing, through functions  $\delta_{\text{call}}$  and  $\delta_{\text{ret}}$ . We deal with them separately. First, removal of  $\delta_{\text{call}}$  can be done by enriching marbles with the current state, so as to pass this information to the assistant transducer. Removal of  $\delta_{\text{ret}}$  is more involved, but can be done by induction using a technique similar to the one used to prove Lemma 34. By induction hypothesis, the assistant transducer can be replaced by an equivalent state-passing free NMT. In addition, its domain is regular thanks to Lemma 34. Hence, one can enrich the marbles so as to precompute the final state reached by the assistant transducer, and in turn simulate the function  $\delta_{\text{ret}}$ . This also allows to ensure that all calls to the assistant transducer do accept.

Last, we justify the fact that the non-elementary blow-up is unavoidable. It is because the domain of any state-passing free NMT  $S$  is recognizable by a finite automaton of exponential size. Indeed, the calls to assistant transducers always terminate, so the domain of  $S$  does not depend on assistant transducers, hence can be described by a marble automaton, hence by a finite automaton of size exponential in the number of local states and marbles of  $S$ . Thus, the existence of an elementary construction for state-passing removal would contradict the non-elementary blow-up stated in Lemma 34. ◀

**Equivalence with lexicographic transductions.** In this subsection, we prove (Theorems 36 and 37) that a transduction is recognizable by a  $k$ -nested marble transducer iff it is  $k$ -lexicographic. Consider some  $f \in \text{LEX}_k$ . Then  $f = \text{maplex}_{\lambda_1}(\text{maplex}_{\lambda_2} \dots \text{maplex}_{\lambda_k} s) \dots$



for some  $\lambda_i = (B_i, \prec_i)$  and  $s$  a simple transduction. We call  $B_1, \dots, B_k$  the ordered alphabets of  $f$  and  $s$  the simple transduction of  $f$ . Given an ordered alphabet  $(B, \prec)$ , one can enumerate the annotations of a word according to the lexicographic extension using a marble automaton.

► **Theorem 36** ( $\text{Lex} \subseteq \text{NMT}$ ). *Any transduction  $f \in \text{Lex}_k$  is recognizable by some NMT  $T_f$  of level  $k$ . If  $B_1, \dots, B_k$  are the ordered alphabets of  $f$  and  $s$  its simple transduction, represented by a sequential transducer with  $m$  states, then  $T_f$  has  $O(k + m)$  states and  $\sum_{i=1}^k |B_i|$  marbles.*

Conversely, we prove that the transductions recognized by  $\text{NMT}_{\text{spf}}$  are lexicographic.

► **Theorem 37** ( $\text{NMT}_{\text{spf}} \subseteq \text{Lex}$ ). *Any transduction  $f$  recognizable by some  $\text{NMT}_{\text{spf}}$   $T$  of level  $k$  is  $k$ -lexicographic.*

The proof is rather involved. We provide high level arguments. The result is shown by induction on  $k$ . It is trivial for  $k = 0$ . For  $k > 0$ , the main idea is to see the sequence of successive configurations of  $T$  on some input as a lexicographic enumeration. This is possible due to the stack discipline of marbles. By extending the marble alphabet with sufficient information, we define a total order on marbles such that successive configurations of  $T$ , extended with this information, forms a lexicographically increasing chain.

## 6 Expressiveness and closure properties of lexicographic transductions

In this section, we prove that  $\text{Lex}$  contains all the polyregular transductions [5], and all the transductions recognizable by (copyful) streaming string transducers [1, 17]. We also show that  $\text{Lex}$  is closed by post-composition under any polyregular transduction, and closed by pre-composition under any rational transduction. We start by showing that lexicographic transductions preserve regular languages under inverse image.

► **Proposition 38.** *Transductions in  $\text{Lex}$  are regularity preserving.*

**Proof.** It is an immediate corollary of the inclusion  $\text{Lex} \subseteq \text{NMT}$  (Theorem 36), that  $\text{NMT}$  are closed by post-composition by a sequential transduction (Lemma 31), and that  $\text{NMT}$  have regular domains (Corollary 35). ◀

We show that  $\text{Lex}$  subsumes both  $\text{SST}$  and  $\text{PolyReg}$ . More precisely that any transduction recognizable by a (copyful) streaming string transducer ( $\text{SST}$ ) is 1-lexicographic. We do not give the definition of  $\text{SST}$  and refer the reader to [12] for more details. We also show that  $\text{NMT}$  of level  $k$  subsume  $k$ -pebble transducers. Again we do not give precise definitions of pebble transducers and refer the reader to [5].

► **Theorem 39** ( $\text{SST}$  and  $\text{PolyReg}$  in  $\text{Lex}$ ). *The following hold:*

- $\text{SST} = \text{Lex}_1$ ,
- A transduction definable by a  $k$ -pebble transducer is in  $\text{Lex}_k$ . In particular  $\text{PolyReg} \subseteq \text{Lex}$ .

**Proof.** It is already known that marble transducers (i.e. nested marble transducers of level 1) capture exactly the class of  $\text{SST}$ -recognizable transductions [12]. The result then follows from Theorem 36 and Theorem 37. For the second statement, we note that single pebble can be simulated by one level of marbles, with colors  $\{0, 1\}$  with the restriction that at most one marble can have color 1 per level. ◀

Any polyregular transduction can be expressed as a composition of sequential transductions, square, map and rev [5]. We show that  $\text{Lex}$  is closed by post-composition by these transductions (e.g. for sequential functions it has been shown in Lemma 31). As a consequence we obtain that  $\text{Lex}$  is closed under post-composition by polyregular transductions.

► **Theorem 40.**  $\text{PolyReg} \circ \text{Lex} \subseteq \text{Lex}$ .

Finally, we show that lexicographic transductions are closed under pre-composition by any rational transduction.

► **Theorem 41** ( $\text{Lex} \circ \text{Rat} \subseteq \text{Lex}$ ). *Let  $k \geq 1$ . For any rational transduction  $g : \Gamma^* \rightarrow \Lambda^*$  and any  $k$ -lexicographic transduction  $f : \Lambda^* \rightarrow \Gamma^*$ ,  $f \circ g$  is  $k$ -lexicographic.*

**Proof.** A rational transduction can be decomposed as a letter-to-letter rational transduction, followed by a morphism. Example 20 shows that morphisms are lexicographic. Similar ideas apply inductively to show that  $\text{Lex}$  is closed by pre-composition under morphisms and letter-to-letter rational transductions. ◀

## 7 Discussion

We have introduced lexicographic transductions, a subclass of  $\text{MSOSI}$ , and provided three characterizations: in terms of closure of simple functions by the  $\text{maplex}$  operator, as lexicographic automatic transductions that can be seen as a syntactic restriction of  $\text{MSOSI}$ , and through nested marble transducers. Thanks to these characterizations, this class is shown to subsume both  $\text{PolyReg}$  and  $\text{SST}$ . Moreover, it is actually closed under post-composition by  $\text{PolyReg}$  and thus is regularity preserving, which  $\text{MSOSI}$  is not known to be.

**Open questions on  $\text{MSOSI}$ .** We leave open whether  $\text{MSOSI}$  is regularity preserving. A way to attack the problem is to try to obtain an equivalent automata-based model, as stated in Question 1. However, as we have shown, a positive answer would entail that all automatic  $\omega$ -words have a decidable  $\text{MSO}$  theory, which has been open for almost 20 years.

**Open questions on  $\text{Lex}$ .** We have shown that  $\text{Lex}$  is a rather well-behaved class, however some interesting questions remain open. Although we suspect that  $\text{MSOSI}$  strictly subsumes  $\text{Lex}$ , this remains unproven. For example, we could not show that  $\text{Lex}$  is closed under pre-composition with  $\text{Reg}$ , in particular, it is already unclear whether  $\text{Lex} \circ \text{rev} \subseteq \text{Lex}$  holds.

The equivalence problem is central to transducer theory, but its decidability status is still unknown for  $\text{PolyReg}$  transductions. It is also open for  $\text{Lex}$ , which subsumes  $\text{PolyReg}$ .

One can decide whether a  $\text{Lex}$  transduction is in  $\text{PolyReg}$  (Theorem 10), however we do not know if the  $\text{Lex}$  hierarchy is decidable (already for its first level  $\text{Lex}_1$ ).

**Possible extensions of  $\text{Lex}$ .** While  $\text{Lex}$  has proven to be an interesting class, the zoo of word-to-word transductions with exponential growth is relatively unknown. We propose three possible extensions of  $\text{Lex}$ , in increasing expressiveness, all included in  $\text{MSOSI}$ .

The class  $\text{Lex} \circ \text{Reg}$  may be an interesting class in itself and the first level  $\text{Lex}_1 \circ \text{Reg}$  coincides with the rather natural class of two-way streaming string transducers.

A more general way of extending  $\text{Lex}$  is to generalize the operation  $\text{lex-enum}$  to allow lexicographic orders where the significance of letters is given by an arbitrary  $\text{MSO}$  definable order. This class subsumes  $\text{Lex} \circ \text{Reg}$  however it is not clear that it is regularity preserving.

Another possible generalization is to replace marbles with the so-called invisible pebbles of Engelfriet [14] and define *nested invisible pebble transducers*, where the nested levels can see the pebbles of the previous levels but not the ones of their own. We believe that the state-passing free version can be shown to be still included in  $\text{MSOSI}$  but it is not clear that it is regularity preserving. However, we conjecture that the version with state-passing can recognize non-regular languages, and hence is too expressive.

## References

- 1 Rajeev Alur and Pavol Cerný. Expressiveness of streaming string transducers. In Kamal Lodaya and Meena Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, volume 8 of *LIPICs*, pages 1–12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPICs.FSTTCS.2010.1.
- 2 Vince Bárány. A hierarchy of automatic  $\omega$ -words having a decidable MSO theory. *RAIRO Theor. Informatics Appl.*, 42(3):417–450, 2008. doi:10.1051/ITA:2008008.
- 3 Jean Berstel, Luc Boasson, Olivier Carton, Bruno Petazzoni, and Jean-Eric Pin. Operations preserving regular languages. *Theor. Comput. Sci.*, 354(3):405–420, 2006. doi:10.1016/J.TCS.2005.11.034.
- 4 Valérie Berthé, Toghrul Karimov, Joris Nieuwveld, Joël Ouaknine, Mihir Vahanwala, and James Worrell. The monadic theory of toric words. *Theor. Comput. Sci.*, 1025:114959, 2025. doi:10.1016/J.TCS.2024.114959.
- 5 Mikolaj Bojańczyk. Polyregular functions. *CoRR*, abs/1810.08760, 2018. arXiv:1810.08760.
- 6 Mikolaj Bojańczyk. On the growth rates of polyregular functions. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–13. IEEE, 2023. doi:10.1109/LICS56636.2023.10175808.
- 7 Mikolaj Bojańczyk, Laure Daviaud, and Shankara Narayanan Krishna. Regular and first-order list functions. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 125–134. ACM, 2018. doi:10.1145/3209108.3209163.
- 8 Mikolaj Bojańczyk, Sandra Kiefer, and Nathan Lhote. String-to-string interpretations with polynomial-size output. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 106:1–106:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ICALP.2019.106.
- 9 Mikolaj Bojańczyk and Rafal Stefanski. Single-use automata and transducers for infinite alphabets. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPICs*, pages 113:1–113:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ICALP.2020.113.
- 10 Michaël Cadilhac, Olivier Carton, and Charles Paperman. Continuity of functional transducers: A profinite study of rational functions. *Log. Methods Comput. Sci.*, 16(1), 2020. doi:10.23638/LMCS-16(1:24)2020.
- 11 Thomas Colcombet and Christof Löding. Transforming structures by set interpretations. *Log. Methods Comput. Sci.*, 3(2), 2007. doi:10.2168/LMCS-3(2:4)2007.
- 12 Gaëtan Douéneau-Tabot, Emmanuel Filiot, and Paul Gastin. Register transducers are marble transducers. In Javier Esparza and Daniel Král’, editors, *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPICs*, pages 29:1–29:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.MFCS.2020.29.
- 13 Joost Engelfriet and Hendrik Jan Hoogetboom. Two-way finite state transducers and monadic second-order logic. In Jiri Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP’99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 311–320. Springer, 1999. doi:10.1007/3-540-48523-6\_28.
- 14 Joost Engelfriet, Hendrik Jan Hoogetboom, and Bart Samwel. XML navigation and transformation by tree-walking automata and transducers with visible and invisible pebbles. *Theor. Comput. Sci.*, 850:40–97, 2021. doi:10.1016/J.TCS.2020.10.030.

- 15 Joost Engelfriet, Hendrik Jan Hoogeboom, and Jan-Pascal van Best. Trips on trees. *Acta Cybern.*, 14(1):51–64, 1999. URL: <https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/3510>.
- 16 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.
- 17 Emmanuel Filiot and Pierre-Alain Reynier. Copyful streaming string transducers. *Fundam. Informaticae*, 178(1-2):59–76, 2021. doi:10.3233/FI-2021-1998.
- 18 Emmanuel Filiot, Pierre-Alain Reynier, and Nathan Lhote. Lexicographic transductions of finite words. *CoRR*, abs/2503.01746, 2025. doi:10.48550/arXiv.2503.01746.
- 19 Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Ann. Pure Appl. Log.*, 130(1-3):3–31, 2004. doi:10.1016/J.APAL.2004.01.007.
- 20 Paul Gallot, Nathan Lhote, and Lê Thành Dũng Nguyễn. The structure of polynomial growth for tree automata/transducers and mso set queries, 2025. doi:10.48550/arXiv.2501.10270.
- 21 Tero Harju and Juhani Karhumäki. Finite transducers and rational transductions. In Jean-Éric Pin, editor, *Handbook of Automata Theory*, pages 79–111. European Mathematical Society Publishing House, Zürich, Switzerland, 2021. doi:10.4171/AUTOMATA-1/3.
- 22 Tsutomu Kamimura and Giora Slutzki. Parallel and two-way automata on directed ordered acyclic graphs. *Inf. Control.*, 49(1):10–51, 1981. doi:10.1016/S0019-9958(81)90438-1.
- 23 Anca Muscholl and Gabriele Puppis. The many facets of string transducers (invited talk). In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPICs*, pages 2:1–2:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.STACS.2019.2.
- 24 Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, USA, 2009.