Revisiting the growth of polyregular functions

Output languages, weighted automata and unary inputs

Sandra Kiefer
University of Oxford, United Kingdom

Lê Thành Dũng (Tito) Nguyễn
École normale supérieure de Lyon, France

Cécilia Pradic
Swansea University, United Kingdom

Abstract

Polyregular functions are the class of string-to-string functions definable by pebble transducers (an extension of finite automata) or equivalently by MSO interpretations (a logical formalism). Their output length is bounded by a polynomial in the input length: a function computed by a k-pebble transducer or by a k-dimensional MSO interpretation has growth rate $O(n^k)$.

Bojańczyk has recently shown that the converse holds for MSO interpretations, but not for pebble transducers. We give significantly simplified proofs of those two results, extending the former to first-order interpretations by reduction to an elementary property of $\mathbb{N}$-weighted automata. For any k, we also prove the stronger statement that there is some quadratic polyregular function whose output language differs from that of any k-fold composition of macro tree transducers (and which therefore cannot be computed by any k-pebble transducer).

In the special case of unary input alphabets, we show that k pebbles suffice to compute polyregular functions of growth $O(n^k)$. This is obtained as a corollary of a basis of simple word sequences whose ultimately periodic combinations generate all polyregular functions with unary input. Finally, we study polyregular and polyblind functions between unary alphabets (i.e. integer sequences), as well as their first-order subclasses.

2012 ACM Subject Classification Theory of computation → Transducers

Keywords and phrases integer-valued polynomials, k-iterative languages, macro tree transducers, (comparison-free) polyregular functions

Funding *Lê Thành Dũng (Tito) Nguyễn*: Supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

Acknowledgements We would like to thank Mikołaj Bojańczyk for giving us much of the initial inspiration that led to this paper; Nathan Lhote for suggesting to us the approach of Section 3; and Gaëtan Douéneau-Tabot for stimulating discussions.

Contents

Warning: This paper is unfinished: among other things, Section 5 is still very messy.

1 Introduction

Many works about *transducers* – automata that can produce output string/trees, not just recognize languages – are concerned with:

- either well-known classes of *linearly growing* functions, i.e. $|f(x)| = O(|x|)$ – in the string-to-string case, those are the sequential, rational and regular functions, see e.g. [39];
- or devices with possibly (hyper)exponential growth rates: HDT0L systems [29, 30, 15], macro tree transducers (MTTs) [26], compositions of MTTs (see below)...

A middle ground is occupied by the *polyregular functions*, surveyed in [5], which derive their name from their *polynomial* growth rate. They are the functions computed by string-to-string *pebble transducers*; although they have been around for two decades (starting with [38], see also [25, 24, 18]), it is only in 2018–2019 that Bojańczyk et al. [3, 6] introduced several alternative characterizations. This had led to renewed interest in this robust function class (e.g. [9, 33]), including applications to linguistics [43].

▶ **Example 1.1** (from [5, §6.2]). The *inner squaring* function, defined as

$$\begin{array}{rccll} \texttt{innsq}: & \{a,b,\#\}^* & \to & \{a,b,\#\}^* & \\ & w_0\#\ldots\#w_n & \mapsto & (w_0)^n\#\ldots\#(w_n)^n & (w_0,\ldots,w_n \in \{a,b\}^*) \end{array}$$

(e.g. $\texttt{innsq}(ba\#abba\#b) = baba\#abbaabba\#bb$), is polyregular and $|\texttt{innsq}(w)| = O(|w|^2)$.

A straightforward property of k-pebble[1] transducers is that their growth rate is[2] $O(n^k)$, and this is the best possible bound. Hence the question of *pebble minimization*: if a polyregular function (defined by some ℓ-pebble transducer) has growth $O(n^k)$, is it always computable by some k-pebble transducer? Definitely not, as Bojańczyk recently showed [4]: no number of pebbles suffices to compute all polyregular functions with *quadratic* growth. Moreover, he uses the tools from [4] to show that `innsq` requires 3 pebbles in [5, Theorem 6.3].[3] (That said, it is known that pebble minimization holds in some restricted settings [14].)

The same paper [4] also contains a proof of a positive result: the polyregular functions with growth $O(n^k)$ are exactly those defined by *k-dimensional MSO interpretations* (a logical

[1] We follow the convention of newer papers for two-way string transducers are 1-pebble transducers, while older papers would have called them 0-pebble transducers; thus, $k \geq 1$ here.
[2] This applies to strings; for trees, (output *height*) $= O((\text{input } \textit{size})^k)$ [25, Lemma 7].
[3] The paper [4] proposes a different quadratic example that requires 3 pebbles, called "block squaring".

formalism introduced in [6]; as usual, MSO stands for Monadic Second-Order logic). But this proof is rather technical, and the one for pebble non-minimization even more so.

Contributions (1): simpler proofs In this paper, we first propose faster ways to reprove the results of [4], sometimes obtaining new results along the way:

- In Section 2, we give a short proof that `innsq` requires 3 pebbles. It only depends on a few familiar properties of regular (not polyregular) functions, such as their closure under composition and an elementary pumping lemma for their output languages.
- We prove dimension minimization for *first-order* (FO) interpretations in Section 3, using a strategy that could directly apply *mutatis mutandis* to MSO interpretations (though we chose to treat the FO case because it has not been explicitly proved in the literature). In fact, we use a reduction to a lemma on first-order queries which is quite similar to the one outlined in [5, §6.1]; but then, instead of using the heavy machinery of factorization forests[4] as in [4] to prove our lemma, we explain how to quickly deduce it from a simple known property of $\mathbb{N}$-weighted automata. Let us remark an immediate corollary of our result: if a first-order polyregular function admits an MSO interpretation of dimension k, it also admits an FO interpretation of dimension at most k.
- In Section 4, we construct a sequence $(f_k)_{k\geq 1}$ of quadratic polyregular functions such that f_k requires $k+1$ pebbles. More than that: the output language of f_k differs from that of any k-fold composition of macro tree transducers. (This composition hierarchy is quite canonical and well-studied, see §4.1.) Again, our proof is quite short,[5] and even arguably easier to check than our ad-hoc argument for `innsq`, though it is less elementary since its "trusted base" is larger: we use a powerful "bridge theorem" on MTTs from [23].

Subclasses of polyregular functions Some restrictions on pebble transducers ensure that:

- the computed function sits at a low level of the aforementioned composition hierarchy;
- a pebble minimization property holds.

This is for instance the case for *k-marble transducers* – they can compute precisely the same string-to-string functions as MTTs[6] with growth[7] $O(n^k)$ [15, §5] – or for *blind pebble transducers*, which define *polyblind*[8] *functions* (cf. Theorem 2.8, taken from [40]). The present paper makes a small contribution to the study of polyblind functions by refuting their conjectured logical characterization (Corollary 3.6; the counterexample is the aforementioned inner squaring). Furthermore, Douéneau-Tabot has recently proved pebble minimization for "last pebble" transducers [14], a model that subsumes both marble and blind pebble transducers; and a function computed by any such device can be obtained by a composition of *two* MTTs (a consequence of [21, Theorem 53 (in §15)]).

---

[4] The factorization forest theorem admits a slight weakening that applies to the aperiodic case. This should enable in principle an adaptation of the proof of [4] to FO interpretations, so we do not claim that our ability to handle this case is a technical advantage of our approach. However, by avoiding factorization forests, our proof strategy should extend to MSO interpretations on trees (see Remark 3.14).

[5] And mostly unoriginal: it consists of little adjustments to an argument by Engelfriet & Maneth [24, §4]. In a followup paper [18], Engelfriet mentions and corrects a mistake in [24, §3], but it does not affect the section which is relevant for our purposes.

[6] Using the fact that string-to-string MTTs are syntactically isomorphic, up to insignificant details, to copyful streaming string transducers.

[7] Interestingly, the proof of marble minimization uses the same property of $\mathbb{N}$-automata that we apply to dimension minimization of FO interpretations.

[8] A name given by Douéneau-Tabot [12, 13, 14] to what Nguyễn, Noûs and Pradic [40] originally called the "comparison-free" subclass of polyregular functions.

Tito — see also [49]

Contributions (2): unary inputs For general polyregular functions with a *unary output alphabet*, pebble minimization also works, because every such function can be computed by a marble transducer [12, §4] (see also Theorem 5.20). What about unary *inputs* (with arbitrary outputs)? Section 5 focuses on this case, that is, on polyregular *word sequences* $\mathbb{N} \to \Sigma^*$ (interpreting input words as unary numerals). We establish pebble minimization in this special case by showing that polyregular sequences[9] can be expressed as "ultimately periodic combinations" of word sequences generated by some simple operations. We believe this latter characterization might be of independent interest: it is inspired by an analogous property of polyblind word sequences [40, §9] that has been applied to prove a separation result [40, Theorem 8.3(i)].

Finally, still in Section 5, we give simple characterizations for polyregular and polyblind *integer sequences* (unary input and output), as well as their first-order (i.e. aperiodic [48]) subclasses. This leads us to a somewhat surprising observation: $n \mapsto n(n-1)/2$ is both polyblind and first-order polyregular, but not first-order polyblind.

Organization of the paper A plan of the sections has already been outlined above; each of them comes with its own preliminaries subsections. Sections 2 and 3 are independent from each other, since they use different characterizations of polyregular functions, while Section 4 reuses some of the preliminaries (but not the new results) of Section 3. Section 5 is mostly independent from the rest of the paper.

Notations. The set of natural numbers is $\mathbb{N} = \{0, 1, \dots\}$. Alphabets are always finite sets.

Let Σ be an alphabet. We write ε for the empty string and Σ^* for the set of strings (or words) with letters in Σ, i.e. the free monoid over Σ; the Kleene star $(-)^*$ will also be applied to languages $L \subseteq \Sigma^*$ as part of usual regular expression syntax. We denote by $\underline{\Sigma}$ a disjoint copy of Σ, made of "underlined letters", so that $a \in \Sigma \mapsto \underline{a} \in \underline{\Sigma}$ is a bijection.

Let $w = w_1 \dots w_n \in \Sigma^*$ ($w_i \in \Sigma$ for $i \in \{1, \dots, n\}$). We use the notations $w[i] = w_i$ and $w[i \dots j] = w_i \dots w_j$ for $1 \leq i \leq j \leq n$. The length $|w|$ of w is n, and $|w|_c$ refers to the number of occurrences of $c \in \Sigma$ in w. We also write $w \not\downarrow i$ for $w_1 \dots w_{i-1} \underline{w_i} w_{i+1} \dots w_n \in (\Sigma \cup \underline{\Sigma})^*$.

The *output language* of a function $f \colon X \to \Sigma^*$ is $f(X) \subseteq \Sigma^*$, also denoted by $\mathrm{Im}(f)$.

In sections 2 and 3, all our (poly)regular functions map ε to ε. This convention goes against the usual practice of allowing the image of the empty string to be any string over the output alphabet. However, we make this choice to avoid inessential inconveniences that arise when dealing with ε in origin semantics and FO interpretations. This makes no difference concerning the strength of our results, since ε could always be dealt with as a special case.

2 A simple proof that inner squaring requires 3 pebbles

While the inner squaring function (Example 1.1) has quadratic growth, we reprove that:

▶ **Theorem 2.1** ([5, Theorem 6.3]). $\mathtt{innsq} \in \mathsf{Pebble}_3 \setminus \mathsf{Pebble}_2$.

Note however that the restriction of `innsq` to inputs in $\{a, \#\}^*$ is in Pebble_2.

[9] In fact, our results apply to a notion of polyregular function $\mathbb{N}^d \to \Sigma^*$.

▶ Remark 2.2. Bojańczyk's proof sketch for [5, Theorem 6.3] actually shows that the function $a_1 \dots a_n \in \mathbb{A}^* \mapsto (a_1)^n \dots (a_n)^n$ cannot be computed by a 2-pebble *atom-oblivious* transducer (here the alphabet $\mathbb{A}$ is an *infinite* set of *atoms*). Combining this with the Deatomization Theorem from [4] – whose proof is rather complicated – shows that no $f\colon \{\langle,\rangle,\bullet\}^* \to \{\langle,\rangle,\bullet\}^*$ in Pebble_2 can satisfy $f(\langle \bullet^{p_1}\rangle \dots \langle \bullet^{p_n}\rangle) = (\langle \bullet^{p_1}\rangle)^n \dots (\langle \bullet^{p_n}\rangle)^n$, from which Theorem 2.1 can easily be deduced using the composition properties of pebble transducers [18].

Instead of introducing the hierarchy $(\mathsf{Pebble}_k)_{k\in\mathbb{N}}$ by a concrete machine model (k-pebble transducers), Section 2.2 defines it using combinators (operators on functions). This abstract presentation depends on the *origin semantics* [2] (see also [39, §5]) of regular functions, about which we say a few words in Section 2.1. We also recall in §2.2 a similar definition of polyblind functions; their only use in this section will be to state and prove Corollary 2.9, but they will appear again in later sections. After all this, Section 2.3 proves Theorem 2.1.

2.1 Regular functions with origin information

We will avoid explicit manipulations of automata computing regular functions. For our purposes here, we just need to know that they are the functions computed by *two-way transducers* (2DFTs), since we use an old result (Lemma 2.13) from the literature on 2DFTs to prove Theorem 2.1. For more on regular functions and their origin semantics, we refer to the survey [39]. A typical example of a regular string-to-string function is:

$$a^{m_0}\#\dots\#a^{m_n} \in \{a,\#\}^* \mapsto a^{m_0}b^{m_0}\#\dots\#a^{m_n}b^{m_n}$$

There are several natural ways to lift it to a regular function with origins, which reflect different ways to compute it with a 2DFT: for example $aaa\#aa$ could be mapped to

$$\begin{array}{cccc} a & a & a & b & b & b & \# & a & a & b & b \\ 1 & 2 & 3 & 1 & 2 & 3 & 4 & 5 & 6 & 5 & 6 \end{array} \quad \text{or} \quad \begin{array}{cccc} a & a & a & b & b & b & \# & a & a & b & b \\ 1 & 2 & 3 & 3 & 2 & 1 & 4 & 5 & 6 & 6 & 5 \end{array}$$

The second component indicates, for each output letter, which input position it "comes from".

We shall write $f^\circ, g^\circ, \dots : \Gamma^* \to (\Sigma \times \mathbb{N})^*$ for regular functions with origin information and $f, g, \dots : \Gamma^* \to \Sigma^*$ for the corresponding regular string-to-string functions. Thus, we have $f = (\pi_1)^* \circ f^\circ$ and $(\pi_2)^* \circ f(w) \in \{1, \dots, |w|\}^*$ for any $w \in \Gamma^*$, where π_i is the projection on the i-th component of a pair ($i \in \{1,2\}$).

2.2 Pebble transducers in an abstract style and polyblind functions

Recall the notations $\underline{\Gamma}$ and $w\lightning i$ from the end of the introduction.

▶ **Definition 2.3.** *Let $f^\circ\colon \Gamma^* \to (I \times \mathbb{N})^*$ be a regular function with origins (so $f : \Gamma^* \to I^*$ is regular) and, for $i \in I$, let $g_i\colon (\Gamma \cup \underline{\Gamma})^* \to \Sigma^*$ and $h_i\colon \Gamma^* \to \Sigma^*$. For $w \in \Gamma^*$, define*

$$\mathsf{pebble}(f^\circ, (g_i)_{i\in I})(w) = g_{i_1}(w\lightning j_1)\cdot \dots \cdot g_{i_n}(w\lightning j_n) \quad \text{where} \quad f^\circ(w) = (i_1,j_1)\dots(i_n,j_n)$$
$$\mathsf{blind}(f, (h_i)_{i\in I})(w) = h_{i_1}(w)\cdot \dots \cdot h_{i_n}(w)$$

Using these combinators, we define the hierarchies of function classes Pebble_n and Blind_n inductively. $\mathsf{Pebble}_0 = \mathsf{Blind}_0$ is the class of string-to-string functions with finite range (or equivalently, whose output has bounded length), and

$$\forall n \in \mathbb{N}, \quad \mathsf{Pebble}_{k+1} = \{\mathsf{pebble}(f^\circ, (g_i)) \mid f^\circ \text{ regular}, \ g_i \in \mathsf{Pebble}_k\}$$
$$\mathsf{Blind}_{k+1} = \{\mathsf{blind}(f, (g_i)) \mid f \text{ regular}, \ g_i \in \mathsf{Blind}_k\}$$

The following properties can be proved routinely.

▶ **Proposition 2.4.** Pebble_1 *and* Blind_1 *are both equal to the class of regular functions. Furthermore,* $\mathsf{Blind}_n \subseteq \mathsf{Pebble}_n$ *for all* $n \in \mathbb{N}$*. Finally, the hierarchies are monotone:* $\mathsf{Pebble}_n \subset \mathsf{Pebble}_{n+1}$ *and* $\mathsf{Blind}_n \subset \mathsf{Blind}_{n+1}$ *for all* $n \in \mathbb{N}$*.*

▶ **Example 2.5.** Let $f^\circ \colon \{a,b,\#\}^* \to (\{\bullet,\#\} \times \mathbb{N})^*$ be defined by

$$f^\circ(\underbrace{w[1]\dots w[i_1-1]}_{\text{each block is in } \{a,b\}^*}\#\dots\#w[i_m+1]\dots w[n]) = \underbrace{\begin{matrix}\bullet & \# & & \bullet & & & \# & \bullet\\ 1 & i_1 & \underbrace{i_1+1}_{\text{each } \bullet \text{ is omitted if the corresponding input block is empty}} & & \cdots & & i_m & i_m+1\end{matrix}}$$

and $h(\dots \#\underline{c}w\# \dots) = cw$ for $c \in \{a,b\}$ and $w \in \{a,b\}^*$. Then

$$\underbrace{\mathtt{innsq}}_{\text{Example 1.1}} = \underbrace{\mathsf{pebble}\big(\overbrace{f^\circ}^{\text{regular with origin information}}, (g_i)_{i\in\{\bullet,\#\}}\big)}_{\in \mathsf{Pebble}_3} \qquad g_\bullet = \underbrace{\mathsf{blind}\big(\overbrace{w \mapsto \bullet^{|w|_\#}}^{\text{both regular i.e. in } \mathsf{Pebble}_1}, (h)_{j\in\{\bullet\}}\big)}_{\in \mathsf{Blind}_2 \subset \mathsf{Pebble}_2},\ g_\# \colon \underbrace{w \mapsto \#}_{\in \mathsf{Pebble}_0}$$

For $k \geq 1$, the class Blind_k corresponds *by definition* to the polyblind functions of rank at most[10] k. Indeed, the combinator blind is precisely the "composition by substitution" operation used to define polyblind functions in [40, §4]. It is then proved in [40, §5] that this coincides with the expressive power of a machine model, the comparison-free (or blind [12, 13, 14]) k-pebble transducers. Similarly, we have:

▶ **Proposition 2.6.** Pebble_k *is exactly the class of string-to-string functions computed by* k*-pebble transducers [5, §2] for any* $k \geq 1$*.*

Proof idea. By a straightforward adaptation of the proof of [40, Corollary 5.7]; therefore, we refer the reader to the long version of [40] for details. To translate Pebble_k into transducers, observe that $\mathsf{pebble}(f^\circ, (g_i)_{i\in I})$ is definable by a k-pebble transducer whenever some two-way transducer (2DFT) computes f° and each g_i is computed by some $(k-1)$-pebble transducer. The converse relies on the fact that the functions computed by 2DFTs *with regular lookaround* are regular [20, Lemma 4]. Note that a similar "origin-based" definition of "last pebble" transducers is stated in [14, Definition 3.3], without explicit proof that they indeed coincide with the expected machine model. ◀

Tito — Actually we should probably give the proof in an appendix or something, and also prove the thing said to be tedious in the following remark

▶ Remark 2.7. By replacing "regular" by "first-order regular" in the definition of Pebble_k, we would get a natural definition of the FO k-pebble hierarchy. However, it is less immediate than for the above proposition to relate it to the first-order k-pebble transducers as defined in [5, Theorem 7.1(2)]. While we expect that the two should indeed coincide, and that this should not involve any conceptual difficulty, we did not attempt a rigorous proof since it would probably be tedious. That said, we shall characterize in Section 5.1 the FO polyregular functions using a similarly abstracted version of *for-transducers* [5, §1].

Finally, let us recall that unlike general pebble transducers (as we shall see soon), blind pebble transducers enjoy the pebble minimization property [40, Theorem 7.1]. For an *effective* minimization proof and a generalization to "last pebble" transducers, see [14].

[10] Actually, in [40, Definition 4.6] they are called the comparison-free polyregular functions of rank at most $k-1$, so there is an offset of 1 with respect to our numbering of the hierarchy.

▶ **Theorem 2.8** ([40, 14]). *For all* $k \in \mathbb{N}$, $\mathsf{Blind}_k = \left\{ f \in \bigcup_{\ell \in \mathbb{N}} \mathsf{Blind}_\ell \;\middle|\; |f(w)| = O(|w|^k) \right\}$.

▶ **Corollary 2.9.** *The inner squaring function (Example 1.1) is not polyblind.*

Proof. Since it has quadratic growth, if it were polyblind, it would be in $\mathsf{Blind}_2 \subset \mathsf{Pebble}_2$. This would contradict the main result of this section (Theorem 2.1). ◀

2.3 Proof of Theorem 2.1

We have seen in Example 2.5 that $\texttt{innsq} \in \mathsf{Pebble}_3$. Our approach to prove $\texttt{innsq} \notin \mathsf{Pebble}_2$ goes through the output languages of regular functions.

▶ **Definition 2.10.** *Call a language* $L \subseteq \Sigma^*$ *a* regular image *if there exists a regular function* f *with codomain* Σ^* *such that* $L = \mathrm{Im}(f)$.

Not all regular images are regular languages, or even context-free languages: consider for instance $\{a^n b^n c^n d^n \mid n \in \mathbb{N}\}$. The usual closure properties of regular functions entail the following for their images:

▶ **Proposition 2.11.** *The class of regular images contains all regular languages* L *with* $\varepsilon \in L$. *If* f *is a regular function and* L, L' *are regular images, so are* $f(L)$ *and* $L \cup L'$.

We will show that no function in Pebble_2 can coincide with $\texttt{innsq}$ on the subset of inputs

$$(a^*b\#)^*\#^* = \{\text{strings with the shape } a \dots ab\# \dots \#a \dots ab\#\# \dots \#\}$$

For the sake of contradiction, assume the opposite.

▷ Claim 2.12. Under this assumption, there exists a language $L \subseteq b\{a,b\}^*b$

- which is a regular image
- that contains only factors of words from $\texttt{innsq}((a^*b\#)^*\#^*)$
- and such that, for any $N \in \mathbb{N}$, some word in L has the shape

$$\overbrace{ba \dots ab \dots b\underbrace{a \dots a}b}^{\text{there are at least } N \text{ } b\text{s}}$$
$$\text{every block of } a\text{s has the same length } n \geq N$$

Proof. We start by an analysis of some output factors arising in the computation of $\texttt{innsq}$ by a hypothetical 2-pebble transducer, which will later inform the construction of a suitable language L. Unfolding our formal definition of Pebble_2, our assumption means that $\texttt{innsq}$ coincides on $(a^*b\#)^*\#^*$ with some function $\mathsf{pebble}(g^\circ, (h_i)_{i\in I})$ where

- $g^\circ \colon \{a,b,\#\}^* \to (\{a,b,\#\} \times \mathbb{N})^*$ is a regular function with origins;
- for $i \in I$, the function $h_i \colon \{a,b,\#,\underline{a},\underline{b},\underline{\#}\}^* \to \{a,b,\#\}^*$ is in Pebble_1, so it is regular.

The functions g° and h_i, being regular, have linear growth: for some $C \in \mathbb{N}$, we have $|g^\circ(u)| \leq C|u|$ and $|h_i(v)| \leq C|v|$ for all nonempty input words u, v and $i \in I$.

Consider the input word $u = (a^n b\#)^n \#^{nm}$ for some $n, m \in \mathbb{N}$ (which shall be taken large enough to satisfy constraints that will arise during the proof). By definition,

$$\texttt{innsq}(u) = \underbrace{h_{i_1}(u \not\downarrow j_1)}_{:=v_1} \cdot \ldots \cdot \underbrace{h_{i_k}(u \not\downarrow j_k)}_{:=v_k} \quad \text{where} \quad g^\circ(w) = (i_1, j_1) \dots (i_k, j_k)$$

For large enough n and m, we have that for all $\ell \in \{1, \dots, k\}$,

$$|v_\ell| \leq C|u \lightning j_\ell| = Cn(n+2+m) < (n+1)n(1+m) = \underbrace{|a^n b| \times |u|_\#}_{\text{distance between two consecutive non-adjacent occurrences of } \# \text{ in } \texttt{innsq}(u)}$$

Thanks to the above, if we call p the largest integer such that $|v_1 \dots v_p|_\# < n$, then:

- $|v_\ell|_\# \leq 1$ for every $\ell \in \{1, \dots, p\}$;
- $v_1 \dots v_p$ is a prefix of $\texttt{innsq}(u)$ that contains all the $n-1$ first occurrences of $\#$.

As a consequence of the latter item, $|v_1 \dots v_p|_b \geq |((a^n b)^{n(1+m)})\#)^{n-1}|$. Meanwhile, we also have $p < k = |g^\circ(u)| \leq C|u|$. Hence

$$\frac{|v_1 \dots v_p|_b}{p} \geq \frac{n(1+m)(n-1)}{Cn(n+2+m)} \xrightarrow[n,m \to +\infty]{} +\infty$$

Thus, by the pigeonhole principle, for an input u parameterized by large enough values of n and m, there is at least one output factor v_ℓ containing $2N$ or more occurrences of b. Since $|v_\ell|_\# \leq 1$, either its largest $\#$-free prefix or its largest $\#$-free suffix (or both) contains N occurrences of b. After applying the regular function $a \dots abwba \dots a \mapsto bwb$ (for $w \in \{a, b\}^*$) to trim this prefix or suffix, we get a factor of v_ℓ – and therefore of $\texttt{innsq}(u)$ – in $b(a^n b)^*$, and we may take $n \geq N$ to ensure that the blocks of as have length at least N.

These observations justify the following definition: let L consist of all words that can be obtained by trimming (in the above sense) the largest $\#$-free prefix or suffix of some $h_i(u \lightning j)$ where $u \in (a^*b\#)^*\#^*$ and (i, j) appears in $g^\circ(u)$. We just argued above that for every N, there are some $n, r \geq N$ such that $b(a^n b)^r \in L$. To conclude, the closure properties of Proposition 2.11 can be used to show that L is a regular image. ◁

Tito — Perhaps expand that last sentence to give more details (also in Proposition 2.11), because it does not have a proof

We shall now use the following old pumping lemma (see also Remark 2.14):

▶ **Lemma 2.13** (Rozoy [44, §4.1]). *If L is a regular image, then for some $k, K \in \mathbb{N}$, every $w \in L$ with $|w| \geq K$ has a decomposition $w = u_0 v_1 \dots u_{k-1} v_k u_k$ with*

- $\exists i \in \{1, \dots, k\}$: $v_i \neq \varepsilon$
- $\forall i \in \{1, \dots, k\}$, $|v_i| \leq K$
- $\{u_0 (v_1)^n \dots u_{k-1} (v_k)^n u_k \mid n \in \mathbb{N}\} \subseteq L$

Let us apply this lemma to the language L given by Claim 2.12, yielding two constants $k, K \in \mathbb{N}$. One of the properties stated in Claim 2.12 is that there exist $n \geq K$ and $r \geq 2k+1$ such that $b(a^n b)^r \in L$. This string has length greater than K, so it is pumpable:

$$b(a^n b)^r = u_0 v_1 \dots u_{k-1} v_k u_k \qquad \text{and} \qquad w = u_0 (v_1)^2 \dots u_{k-1} (v_k)^2 u_k \in L$$

Let us now show that $ba^n b$ is a factor of w. Since $|v_i| \leq K \leq n$ for every $i \in \{1, \dots, k\}$, and any two occurrences of b in $b(a^n b)^r$ are separated by a^n, each factor v_i can contain at most one b. So $|v_1|_b + \dots + |v_k|_b \leq k$ which leads to $|u_0|_b + \dots + |u_k|_b \geq r + 1 - k \geq k + 2$. One of the u_j must then contain two occurrences of b, and this u_j is also a factor of w.

Recall that L is comprised exclusively of $\#$-free factors of words in $\texttt{innsq}((a^*b\#)^*\#^*)$, so *all the "blocks of as" in w must have the same size.* Having seen above that this size must be n, we can now conclude by a case analysis, reaching a contradiction in both cases:

- First assume that $|v_i|_b \geq 1$ for some $i \in \{1, \dots, k\}$. We may write $v_i = a^\ell b \dots b a^m$ (where the first and last b can coincide) in this case. Then w contains $(v_i)^2$ which in turn contains a factor $ba^{\ell+m}b$. So $\ell + m = n$ since it is the size of a block of as in w; but at the same time, using the second item of Lemma 2.13, $\ell + m < |v_i| \leq K \leq n$.
- Otherwise, $v_i \in \{a\}^*$ for all i, so pumping does not increase the number of bs. Therefore w has as many blocks of as as $b(a^n b)^r$, and we have also seen that its blocks have size n, so $w = b(a^n b)^r$. But since at least one v_i is nonempty, $|w| > |b(a^n b)^r|$.

▶ Remark 2.14. Rozoy proves Lemma 2.13 for two-way transducers (cf. §2.1). She also shows [44, §4.2] that the output languages of *nondeterministic* two-way transducers enjoy a weaker version of the lemma without the bound on the length of the v_i (this bound is refuted by the example $\{(w\#)^n \mid w \in \Sigma^*, n \in \mathbb{N}\}$); in general, the languages that satisfy this weaker pumping lemma are called *k-iterative* in the literature (see e.g. [47, 34]). For more on regular images, see [16, 19]; several references are also given in [32, p. 18].[11]

3 Dimension minimization for first-order interpretations

We first recall the definition of FO interpretations in Section 3.1 and give several examples. Then Section 3.2 states the dimension minimization theorem, and explains how to derive it as a consequence of a lemma on queries defined by first-order formulas. To establish this lemma, we first recall some background on $\mathbb{N}$-weighted automata in §3.3 and then use it to give a proof in §3.4.

3.1 First-order interpretations

We assume basic familiarity with first-order logic (FO).

Let Σ be an alphabet. A word $w \in \Sigma^*$ can be seen as a structure whose domain is the set of *positions* $\{1, \dots, |w|\}$, over the relational vocabulary consisting of:

- for each $c \in \Sigma$, a unary symbol c where $c(i)$ is interpreted as true whenever $w[i] = c$,
- and a binary relation symbol $\leq$, interpreted as the total order on positions.

As an example, for $\Sigma = \{a, b\}$, let $F(x) = b(x) \vee \forall y.\ (a(y) \vee x \leq y)$. A string $w \in \Sigma^*$ satisfies $F(i)$ for a given $i \in \{1, \dots, w\}$ – notation: $w \models F(i)$ – when either $w[i] = b$, or the position i contains an a which occurs before (at the left of) all the bs in the word w. Thus, the formula $\forall x.\ F(x)$ evaluates to true exactly over the words in a^*b^*. (It is precisely the star-free languages that can be defined this way, see [48] for a history of this seminal result.)

This model-theoretic perspective also leads to a way to specify string-to-string functions. (Note that FO interpretations make sense for – and have been intensively studied in the context of – more general relational structures such as graphs, see e.g. [10, 11].)

▶ **Definition 3.1** (variant of [5, §5.1][12]). *Let $k \geq 1$ and Γ, Σ be alphabets. A k-dimensional first-order interpretation $\mathcal{I}$ from Γ^* to Σ^* consists of*

- *a finite set of* components $\mathcal{I}_{\text{comp}}$*;*
- *an FO formula $\mathcal{I}_c^\alpha(x_1, \dots, x_k)$ over Γ for each $c \in \Sigma$ and $\alpha \in \mathcal{I}_{\text{comp}}$;*

[11] Instead of Latteux's unobtainable technical report cited by [32], see [36, Prop. I.2] & compare with [42].

[12] Bojańczyk's definition in [5, §5.1] uses components with varying arities $\leq k$. Those are essentially equivalent to our definition except that the arity 0 components provide a way to encode non-empty outputs for the empty input string – a case that we have excluded by convention (as mentioned at the end of the introduction). The definition used by Bojańczyk, Kiefer & Lhote in [6] is equivalent to our single-component interpretations; but see Remark 3.8.

- *an FO formula* $\mathcal{I}_{\leq}^{\alpha,\beta}(x_1,\dots,x_k,y_1,\dots,y_k)$ *over* Γ *for each* $(\alpha,\beta)\in(\mathcal{I}_{\text{comp}})^2$.

It defines the function $u\in\Gamma^*\mapsto\begin{cases}v & \text{if } O_u\cong \text{the structure corresponding to } v\\ \varepsilon & \text{when}^{13} \text{ there is no such } v\in\Sigma^*\end{cases}$

where O_u *is the structure*

- *with domain* $\{\langle\alpha,i_1,\dots,i_k\rangle\in\times\{1,\dots,|u|\}^k \mid \exists c\in\Sigma\colon u\models\mathcal{I}_c^\alpha(i_1,\dots,i_k)\}$;
- *where* $c(\langle\alpha,i_1,\dots,i_k\rangle)$ *iff* $u\models\mathcal{I}_c(i_1,\dots,i_k)$ *for* $c\in\Sigma$ *and* $\alpha\in\mathcal{I}_{\text{comp}}$;
- *where* $\langle\alpha,i_1,\dots,i_k\rangle\leq\langle\beta,j_1,\dots,j_k\rangle$ *iff* $u\models\mathcal{I}_{\leq}^{\alpha,\beta}(i_1,\dots,i_k,j_1,\dots,j_k)$.

$\mathcal{I}$ *is said to be a* single-component *FO interpretation when* $|\mathcal{I}_{\text{comp}}|=1$. *In that case we usually omit the components* $\alpha,\beta,\dots$

▶ Remark 3.2. It follows directly from the definitions that if a function f is defined by a first-order interpretation with N components, then $|f(u)|\leq N|u|^k$.

▶ **Example 3.3.** Consider the following 1-dimensional FO interpretation: $\mathcal{I}_{\text{comp}}=\{\lambda,\rho\}$ and

- $\mathcal{I}_a^\lambda(x)=\mathcal{I}_b^\rho(x)=\texttt{true}$ whereas $\mathcal{I}_b^\lambda(x)=\mathcal{I}_a^\rho(x)=\texttt{false}$
- $\mathcal{I}_{\leq}^\lambda(x,y)=\mathcal{I}_{\leq}^\rho(x,y)=(x\leq y)$ and $\mathcal{I}^{\lambda,\rho}(x,y)=\texttt{true}$ and $\mathcal{I}^{\rho,\lambda}(x,y)=\texttt{false}$

The function that it defines is $w\in\Gamma^*\mapsto a^{|w|}b^{|w|}$. Note that we could get the same function by taking instead $\mathcal{I}_{\leq}^\rho(x,y)=(y\leq x)$ for example.

▶ **Example 3.4.** $a^n\in\{a\}^*\mapsto(a^{n-1}b)^{n-1}$ is defined by a single-component 2-dimensional FO interpretation $\mathcal{I}$ such that $\mathcal{I}_\leq$ is the lexicographic order over pairs and

$$\mathcal{I}_a(x_1,x_2)=\neg\underbrace{\max(x_1)}_{\text{i.e. }\forall y.\ y\leq x_1}\wedge\neg\max(x_2)\qquad\mathcal{I}_b(x_1,x_2)=\neg\max(x_1)\wedge\max(x_2)$$

▶ **Example 3.5.** As a more subtle example, the inner squaring function (Example 1.1) admits a single-component 2-dimensional FO interpretation that we intuitively illustrate as follows:

$$\begin{array}{cccc}
 & baba\#abba\#b & & \\
\vdots & \cdots & & \\
\# & baba\quad abba\quad b & \rightsquigarrow & bababa ba\#abbaabba\#bb \\
\vdots & \cdots & & \\
\# & baba\quad abba\quad b & & \\
b & \#\qquad\# & &
\end{array}$$

- $\mathcal{I}_a(x_1,x_2)=a(x_1)\wedge\#(x_2)$ and $\mathcal{I}_b(x_1,x_2)=b(x_1)\wedge\#(x_2)$
- $\mathcal{I}_\#(x_1,x_2)=\#(x_1)\wedge\forall y_2.\ y_2\leq x_2$
- $\mathcal{I}_\leq(x_1,x_2,y_1,y_2)=$ either $\#(y_1)\wedge x_1\leq y_1$, or there exist x_1,y_1 such that
  - $x_3\leq x_1$, and there are no #s strictly in-between x_3 and x_1;
  - $y_3\leq y_1$, and there are no #s strictly in-between y_3 and y_1;
  - neither x_3 nor y_3 has an immediate predecessor which is an a or a b;
  - $(x_3,x_2,x_1)\leq(y_3,y_2,y_1)$ for the lexicographic order on 3-tuples.

▶ **Corollary 3.6.** *The characterization of polyblind functions conjectured in [40, §10] is wrong.*

[13] This is purely for convenience, to avoid having to distinguish between partial and total functions. The language of input words u for which such a v exists is star-free/FO-definable, so a partial FO interpretation can always be completed to a total one.

Proof. For purely syntactic reasons, the 2-dimensional FO interpretations such that:

- in every formula defining the interpretation, each variable can be given a sort in $\{1, 2\}$ in such a way that variables of different sorts are never compared (for equality or ordering)
- and in $\mathcal{I}(x_1, x_2, y_1, y_2)$ and $\mathcal{I}_c(x_1, x_2)$ ($c \in \Sigma$), x_i and y_i have sort $i \in \{1, 2\}$

can be seen as a special case of the logical interpretations considered in [40, Conjecture 10.1]. Example 3.5 fits these criteria (with x_3 and y_3 having sort 1). If the conjecture were true, `innsq` would therefore be polyblind, contradicting Corollary 2.9. ◀

▶ **Theorem 3.7** (Bojańczyk, Kiefer & Lhote [6])**.** *The functions specified by FO interpretations are exactly the first-order polyregular functions.*

▶ Remark 3.8. In [6] it is explained that for any first-order polyregular function, there is a single-component FO interpretation which coincides with it on every input of length ≥ 2. But dimension minimization requires multiple components.

3.2 Reducing dimension minimization to a lemma on FO queries

We wish to prove a sort of converse to Remark 3.2:

▶ **Theorem 3.9.** *Let f be a first-order polyregular function, and let $k \in \mathbb{N}$ be the minimal exponent such that $|f(w)| = O(|w|^k)$. Then f is specified by some FO interpretation $\mathcal{J}$ of dimension at most k. Moreover, both k and $\mathcal{J}$ can be effectively computed from a given interpretation of possibly non-optimal dimension defining f.*

As mentioned in the introduction, the counterpart to this theorem for MSO interpretations and general polyregular functions has been proved by Bojańczyk [4, §2]. The survey [5] gives a reduction [5, proof of Theorem 6.1] to a "Seed Lemma" [5, Lemma 6.2] concerning arbitrary MSO formulas. We analogously reduce Theorem 3.9 to a similar (but weaker, to make our job easier) lemma on FO formulas.

▶ **Lemma 3.10.** *Let $F(x_1, \dots, x_\ell)$ be a first-order formula over Γ^*. One can compute:*

- *the least $k \in \mathbb{N}$ such that $|\{(i_1, \dots, i_\ell) \mid w \models F(i_1, \dots, i_\ell)\}| = O(|w|^k)$ (so $k \leq \ell$);*
- *a formula $G(x_1, \dots, x_\ell, z_1, \dots, z_k)$ and a bound $B \in \mathbb{N}$ such that for every $w \in \Gamma^*$,*
  - *$\forall j_1, \dots, j_k$, $|\{(i_1, \dots, i_\ell) \mid w \models G(i_1, \dots, i_\ell, j_1, \dots, j_k)\}| \leq B$;*
  - *$\forall i_1, \dots, i_\ell$, $w \models F(i_1, \dots, i_\ell) \implies |\{(j_1, \dots, j_k) \mid w \models G(i_1, \dots, i_\ell, j_1, \dots, j_k)\}| = 1$.*

Proof of Theorem 3.9. Let $\mathcal{I}$ be an ℓ-dimensional FO interpretation from Γ^* to Σ^*. Let

$$F(x_1, \dots, x_\ell) = \bigvee_{\alpha \in \mathcal{I}_{\text{comp}}, c \in \Sigma} \mathcal{I}_c^\alpha(x_1, \dots, x_\ell)$$

By Lemma 3.10, one can compute k, G and B verifying several properties. Note that we have $|\mathcal{I}(w)| = \Theta(|\{(i_1, \dots, i_\ell) \mid w \models F(i_1, \dots, i_\ell)\}|)$ (their ratio is between 1 and $|\mathcal{I}_{\text{comp}}|$); therefore, k is the smallest integer such that $|\mathcal{I}(w)| = O(|w|^k)$.

For $m \in \{1, \dots, B\}$, there is an FO formula $H_m(x_1, \dots, x_\ell, z_1, \dots, z_k)$ whose meaning is: among the tuples $(y_1, \dots, y_\ell)$ such that $G(y_1, \dots, y_\ell, z_1, \dots, z_k)$, the tuple $(x_1, \dots, x_\ell)$ is the m-th one (for some fixed first-order definable total order, e.g. lexicographic). (H_m is false whenever there are strictly less than m such tuples, or $(x_1, \dots, x_\ell)$ is not one of those tuples.) The following k-dimensional interpretation $\mathcal{J}$ then computes the same function as $\mathcal{I}$:

- $\mathcal{J}_{\text{comp}} = \mathcal{I}_{\text{comp}} \times \{1, \dots, B\}$
- $\mathcal{J}_c^{(\alpha,m)}(z_1, \dots, z_k) = \exists x_1. \ \dots \exists x_\ell.\ H_m(x_1, \dots, x_\ell, z_1, \dots, z_k) \wedge \mathcal{I}_c^\alpha(x_1, \dots, x_\ell)$
- $\mathcal{J}_\leq^{(\alpha,m),(\beta,p)}(\vec{z}, \vec{z}\,') = \exists \vec{x}.\ \exists \vec{x}\,'.\ H_m(\vec{x}, \vec{z}) \wedge H_p(\vec{x}\,', \vec{z}\,') \wedge \mathcal{I}_\leq^{\alpha,\beta}(\vec{x}, \vec{x}\,')$. ◀

To conclude our proof, we must show Lemma 3.10. Towards this aim, we now recall some tools used in our approach.

3.3 $\mathbb{N}$-weighted automata

The classical notion of weighted automaton over a semiring is presented for instance in the reference book [45, Chapter III]. We consider only the case where the weights are in $\mathbb{N}$, and follow the notations of [15, §5.2].

▶ **Definition 3.11.** *An $\mathbb{N}$-automaton $\mathcal{A}$ over the alphabet Γ consists of:*

- *a finite set Q of states;*
- *an initial row vector $\mathbf{i} \in \mathbb{N}^Q$ and final column vector $\mathbf{f} \in \mathbb{N}^Q$;*
- *a monoid morphism $\mu \colon \Gamma^* \to \mathbb{N}^{Q \times Q}$ (with matrix multiplication).*

It computes the function $w \in \Gamma^* \mapsto \mathcal{A}(w) = \mathbf{i} \cdot \mu(w) \cdot \mathbf{f} \in \mathbb{N}$.

▷ Claim 3.12. The usual notion of *deterministic finite automaton (DFA)* is equivalent to an $\mathbb{N}$-automaton such that:

- $\mathbf{i}$ has a single coordinate equal to 1, and the others are equal to 0;
- the same property applies to every row of every matrix $\mu(c)$ for $c \in \Gamma$;
- $\mathbf{f} \in \{0,1\}^{\mathbb{N}}$.

This implies $\mathcal{A}(w) \in \{0,1\}$ for every $w \in \Gamma^*$, so $\mathcal{A}$ seen as an $\mathbb{N}$-automaton computes the characteristic function of the regular language recognized by $\mathcal{A}$ seen as an usual DFA.

Throughout the rest of this section we shall therefore identify DFA with $\mathbb{N}$-automata of this shape. We will also use $\mathbb{N}$-automata that take unbounded – indeed, polynomially growing – values. Their main property of interest for us is:

▶ **Lemma 3.13** ([15, §5.2])**.** *Let $\mathcal{A} = (Q, \mathbf{i}, \mathbf{f}, \mu)$ be an $\mathbb{N}$-automaton over Γ. Suppose that:*

- *$\mathcal{A}$ is* trim*, i.e. $\forall q \in Q,\ \exists u, v \in \Gamma^* \colon (\mathbf{i} \cdot \mu(u))(q) \geq 1 \wedge (\mu(v) \cdot \mathbf{f})(q) \geq 1$;*
- *$\mathcal{A}(w)$ grows at most polynomially[14] in $|w|$.*

Then one can compute from $\mathcal{A}$ the smallest $k \in \mathbb{N}$ such that $\mathcal{A}(w) = O(|w|^k)$. Furthermore, there is a computable partition $Q = Q_0 \sqcup \cdots \sqcup Q_k$ of the states such that:

- *$\mu(w)(q, q') = 0$ for any $i > j$, $q \in Q_i$, $q' \in Q_j$ and $w \in \Gamma^*$;*
- *$\mu(w)(q, q') = O(1)$ for any fixed $i \in \{0, \dots, k\}$ and $q, q' \in Q_i$ – the bound depends on $\mathcal{A}$, and is computable.*

▶ Remark 3.14. This is very similar to some results in the literature on *polynomially ambiguous* weighted automata (over arbitrary semirings) such as [35, Theorem 6.2]. But the extensional notion of growth of the function computed by an $\mathbb{N}$-automaton fits better our intended application than the intensional degree of ambiguity of automata.

We also note that polynomially ambiguous *tree* automata admit a decomposition into simpler analyzable parts [41, Theorems 9 and 12] which "morally" seems to entails an extension of Bojańczyk's "Seed Lemma" [5, Lemma 6.2] to trees (though formally we could not deduce the latter from the precise statement given for the former). Thus, we expect that MSO tree-to-tree interpretations should also admit a dimension minimization theorem. It is unclear whether dimension minimization for trees could be proved using the techniques of [4], since factorization forests do not work on trees.

[14] Actually, subexponential growth implies polynomial growth for $\mathbb{N}$-automata, but we will not need this.

3.4 Proof of Lemma 3.10

First recall that the celebrated correspondence between first-order logic and star-free languages (cf. [48]) also extends to formulas with free variables in the following way, that provides a useful "logic-free" perspective on queries defined by first-order formulas:

▶ **Theorem 3.15.** *Let $\ell \in \mathbb{N}$ and $\Xi \subseteq \bigcup_{w\in\Gamma^*} \{w\} \times \{1,\dots,|w|\}^\ell$. The following are equivalent:*

- *there is an FO formula $F(x_1,\dots,x_\ell)$ such that $(w, i_1,\dots,i_\ell) \in \Xi$ iff $w \models F(i_1,\dots,i_\ell)$;*
- *there is a star-free language $L \subseteq (\Gamma \times \{0,1\}^\ell)^*$ such that $(w, i_1,\dots,i_\ell) \in \Xi$ iff L contains the word $\mathsf{marked}(w, i_1,\dots,i_\ell)$ of length $|w|$ whose j-th letter*
  - *has $w[j]$ as its first coordinate;*
  - *has 1 as its $(m+1)$-th coordinate if $i_m = j$, and 0 otherwise, for $m \in \{1,\dots,\ell\}$;*
- *the "language of marked words" $\{\mathsf{marked}(\xi) \mid \xi \in \Xi\}$ is star-free.*

Let $F(x_1,\dots,x_\ell)$ be an FO formula over Γ^*, and let $\mathcal{A} = (Q, \mathbf{i}, \mathbf{f}, \mu)$ be the *minimal* DFA over $(\Gamma \times \{0,1\}^\ell)^*$ (seen as an $\mathbb{N}$-automaton) computing its language of marked words. We define another $\mathbb{N}$-automaton $\widehat{\mathcal{A}} = (Q, \mathbf{i}, \mathbf{f}, \widehat{\mu})$ over Γ^* by taking for each $c \in \Gamma$:

$$\widehat{\mu}(c) = \sum_{\vec{b}\in\{0,1\}^\ell} \mu\Big(\underbrace{(c,\vec{b})}_{\text{a 1-letter word}}\Big) \quad \text{so that} \quad \forall w \in \Gamma^*,\ \widehat{\mathcal{A}}(w) = |\{(i_1,\dots,i_\ell) \mid w \models F(i_1,\dots,i_\ell)\}|$$

Since $\mathcal{A}$ is minimal, it is trim, so $\widehat{\mathcal{A}}$ is also trim. Apply Lemma 3.13 to $\widehat{\mathcal{A}}$ to get $k \in \mathbb{N}$ – the degree of growth of $\widehat{\mathcal{A}}$ and therefore of F – and a partition $Q = Q_0 \sqcup \cdots \sqcup Q_k$. The fact that $\widehat{\mathcal{A}}$ is trim and the conditions on the partition imply that the initial state of $\mathcal{A}$ is in Q_0.

We use this partition to define a new DFA $\widetilde{\mathcal{A}} = (Q, \mathbf{i}, \mathbf{f}, \widetilde{\mu})$ over $(\Gamma \times \{0,1\}^{\ell+k})^*$: for $c \in \Gamma$, $\vec{b} \in \{0,1\}^\ell$, $\vec{b'} \in \{0,1\}^k$, $i, j \in \{0,\dots,k\}$, $q \in Q_i$ and $q' \in Q_j$,

$$\widetilde{\mu}\Big((c,\vec{b},\vec{b'})\Big)(q,q') = \begin{cases} \mu\Big((c,\vec{b})\Big)(q,q') & \text{if } \vec{b'} = (0,\dots,0,\underbrace{1,\dots,1}_{\text{from the } (i+1)\text{-th to the } j\text{-th coordinate – so } \vec{b'}=\vec{0} \text{ if } i=j},0,\dots,0) \\ 0 & \text{otherwise} \end{cases}$$

$\widetilde{\mathcal{A}}$ is a counter-free DFA. Indeed, for any $w \notin (\Gamma \times \{0,1\}^\ell \times \{0\}^k)^*$, the matrix $\widetilde{\mu}(ww)$ is zero (due to the first item in the conclusions of Lemma 3.13), so checking that $\widetilde{\mathcal{A}}$ is counter-free reduces to the fact that $\mathcal{A}$ is counter-free – which is the case since $\mathcal{A}$ is the minimal DFA of a star-free language. So $\widetilde{\mathcal{A}}$ recognizes a star-free language. Because (again) of the first item of Lemma 3.13, this language is of the form $\{\mathsf{marked}(\xi) \mid \xi \in \widetilde{\Xi}\}$ where $\widetilde{\Xi}$ contains tuples of the form $(w, i_1,\dots,i_\ell, j_1,\dots,j_m)$ for $m \in \{0,\dots,k\}$ (the corresponding marked word reaches a final state in Q_m when read by $\widetilde{\mathcal{A}}$). Let

$$\overline{\Xi} = \{(w, i_1,\dots,i_\ell, j_1,\dots,j_m, \underbrace{1,\dots,1}_{k-m \text{ times}}) \mid (w, i_1,\dots,j_m) \in \widetilde{\Xi}\} \subseteq \bigcup_{w\in\Gamma^*} \{w\} \times \{1,\dots,|w|\}^{\ell+k}$$

Then the language of marked words for $\overline{\Xi}$ is still star-free, so $\overline{\Xi}$ corresponds to some first-order formula $G(x_1,\dots,x_\ell,z_1,\dots,z_k)$. Using the second item of Lemma 3.13, one can see that G verifies the conclusions of the Lemma 3.10 that we wanted to prove.

Tito — more details?

4 Quadratic polyregular functions vs macro tree transducers

Our goal now is to show the following:

▶ **Theorem 4.1.** *For any $k \geq 1$, there exists a string-to-string function f_k such that:*

- *f_k is computed by a* single-component 2-dimensional first-order interpretation *(cf. §3.1);*
- *for any k-tuple of functions $(g_1, \dots, g_k)$ such that each g_i is computed by some* macro tree transducer*, $\mathrm{Im}(g_1 \circ \cdots \circ g_k) \neq \mathrm{Im}(f_k)$.*

In the second item, the domain of g_i must be equal to the codomain of g_{i+1} for the composition to be well-defined; and to make the comparison of output languages meaningful, the codomains of g_1 and f_k should be equal.

But g_1 outputs ranked trees, whereas f_1 outputs strings. To make sense of this, as usual, we identify Σ^* with the set of trees over the ranked alphabet that consists of a letter $\widehat{c}$ of rank 1 for each $c \in \Sigma$, plus a single letter $\widehat{\varepsilon}$ of rank 0. Through this identification, the domain of g_k and f_k may also be equal; in that case we may conclude that $g_1 \circ \cdots \circ g_k \neq f_k$.

The relevance of Theorem 4.1 to the question of pebble minimization comes from:

▶ **Theorem 4.2** (Engelfriet & Maneth [25, item (2) of the abstract])**.** *Any tree-to-tree function computed by some k-pebble*[15] *tree transducer can also be expressed as a k-fold composition of macro tree transducers. (But the converse is false.)*

A k-pebble string transducer is none other than a k-pebble tree transducer working on the encodings of strings as unary trees described above. Thus it follows directly that $f_k \notin \mathsf{Pebble}_k$ (without having to recall the definition of Pebble_k from §2.2), and even that $\mathrm{Im}(f_k) \neq \mathrm{Im}(h)$ for any $h \in \mathsf{Pebble}_k$.

On the other hand, any 2-dimensional FO interpretation is polyregular (Theorem 3.7) with quadratic growth (Remark 3.2). So the f_k are indeed counterexamples to pebble minimization.

▶ Remark 4.3. Our main reason for using first-order interpretations is that they provide a nice notion of multidimensional origin semantics, which we can use in our inductive construction of the f_k. But we do not truly depend on the difficult translation of FO interpretations to pebble transducers [6] to refute pebble minimization: it would be possible to show by a simpler ad-hoc argument that each f_k is computable by some ℓ-pebble transducer.

We recall the required properties of macro tree transducers in Section 4.1, then we prove Theorem 4.1 in Section 4.2.

4.1 Compositions of macro tree transducers (MTTs)

We do not formally define MTTs here, but only recall useful facts for our purposes. In this paper, we only consider *total deterministic* MTTs. Let MTT^k be the class of tree-to-tree functions computed by some composition of $k \geq 1$ macro tree transducers.

▶ Remark 4.4. MTT^k can be characterized equivalently as the class of functions computed by k-iterated pushdown transducers [27], or by "level-k" tree transducers [28]. For any k, the functions in MTT^k with linear growth are precisely the regular tree functions [22][16] –

[15] As explained in Footnote 1 of the introduction, the indexing convention for the pebble transducer hierarchy in [25] is off by one compared to ours.

[16] The paper [22] talks about compositions of tree-walking transducers (TWT), but MTT^1 is included in the class of functions obtained by composing 3 TWTs [25, Lemma 37].

combined with Theorem 4.2, this proves pebble minimization for polyregular functions of linear growth,[17] so our counterexamples with quadratic growth are in some sense minimal.

For a class of tree-to-tree functions $\mathcal{C}$, we also write $\mathrm{SO}(\mathcal{C})$ for the subclass of functions that output strings (via the identification described at the beginning of §4). The literature on tree transducers often refers to the classes $\mathrm{SO}(\mathsf{MTT}^k)$ by equivalent descriptions involving a *yield* operation[18] that maps trees to strings:

▷ Claim 4.5. $\mathrm{SO}(\mathsf{MTT}^{k+1}) = \{\mathtt{yield} \circ f \mid f \in \mathsf{MTT}^k\}$ for any $k \geq 1$, whereas $\mathrm{SO}(\mathsf{MTT}^1)$ consists of the functions $\mathtt{yield} \circ$ (some *top-down* tree transducer).

Explanation. The case $k = 1$ of the first claim – which directly entails the general case – is a consequence of the work of Engelfriet and Vogler [27, items (a) + (d) of the introduction]. To understand why requires a terminological clarification: while, according to old definitions,

$$X \text{ tree-to-string transducer} = \mathtt{yield} \circ (X \text{ tree transducer}) \text{ for } X \in \{\text{macro}, \text{top-down}\}$$

(for the top-down case see e.g. [37, 46]), this does not apply to $X = \text{pushdown}^n$. Actually, pushdown^n tree-to-string transducers $= \mathrm{SO}(\text{pushdown}^n$ tree transducers) in [27].

The claim about $\mathrm{SO}(\mathsf{MTT}^1)$ is well-known. For instance it is stated by Maneth as follows: "Note that macro tree transducers with monadic output alphabet are essentially the same as top-down tree-to-string transducers" [37, end of §5]. ◁

This allows us to rephrase a key "bridge theorem" by Engelfriet and Maneth [23] in a form that suits us better. For a class of string-valued functions $\mathcal{C}'$, let $\mathrm{Im}(\mathcal{C}') = \{\mathrm{Im}(f) \mid f \in \mathcal{C}'\}$.

▶ **Theorem 4.6** ([23, Theorem 18] + Claim 4.5)**.** *Let $k \geq 1$ and L, L' be string languages. If L' is* d-complete (cf. below) *for L and $L' \in \mathrm{Im}(\mathrm{SO}(\mathsf{MTT}^{k+1}))$, then $L \in \mathrm{Im}(\mathrm{SO}(\mathsf{MTT}^k))$.*

▶ **Definition 4.7.** *Let $L \subseteq \Sigma^*$ and $L' \subseteq (\Sigma \cup \Delta)^*$ with $\Sigma \cap \Delta = \varnothing$. We say that L' is:*

- *δ-complete for L [23, §5] when for every $u \in L$ there exist $w_0, \ldots, w_n \in \Delta^*$ such that*
  - *all the w_i for $i \in \{1, \ldots, n-1\}$ are* pairwise distinct *words;*
  - *$n = |u|$ and $w_0 u[1] w_1 \ldots u[n] w_n \in L'$;*
- d-complete for L [24, §4] *when it is δ-complete for L and "conversely", by erasing the letters from Δ in the words in L' one gets exactly L, i.e. $\varphi(L') = L$ where φ is the monoid morphism mapping each letter in Σ to itself and Δ to ε.*

4.2 Proof of Theorem 4.1

Our strategy is a minor adaptation of Engelfriet and Maneth's proof [24, §4] that the hierarchy of output languages of k-pebble string transducers is strict. (See also [25, Theorem 41] for a similar result on tree languages.) Writing $\mathrm{Im}(\mathcal{I}) = \mathrm{Im}(f)$ when the first-order interpretation $\mathcal{I}$ defines the string function f, we show that:

▶ **Lemma 4.8.** *From any single-component 2D FO interpretation $\mathcal{I}$, one can build another single-component 2D FO interpretation $\Psi(\mathcal{I})$ such that $\mathrm{Im}(\Psi(\mathcal{I}))$ is d-complete for $\mathrm{Im}(\mathcal{I})$.*

▶ Remark 4.9. Our construction preserves the class of interpretations that verify the property described in the proof of Corollary 3.6.

[17] This special case is also a consequence of dimension minimization for MSO interpretations.
[18] $\mathtt{yield}(t)$ is the word formed by listing the labels of the leaves of t in infix order.

Before proving the lemma, we explain how it leads to Theorem 4.1. For $k \geq 1$, let

$$\mathcal{I}_k = \Psi^{k-1}(\text{the FO interpretation of Example 3.4})$$

$\mathcal{I}_1$ maps a^n to $(a^{n-1}b)^{n-1}$ for $n \geq 1$, so we have[19] $\mathrm{Im}(\mathcal{I}_1) \notin \mathrm{Im}(\mathrm{SO}(\mathsf{MTT}^1))$ by immediate application of an old result of Engelfriet [17, Theorem 3.16].[20] Together, Lemma 4.8 and the contrapositive of Theorem 4.6 entail that $\mathrm{Im}(\mathcal{I}_k) \notin \mathrm{Im}(\mathrm{SO}(\mathsf{MTT}^k))$ for all k – which is precisely what we want.

Proof of Lemma 4.8. The idea is inspired by the coding of atom-oblivious functions used in the Deatomization Theorem of [4] (cf. Remark 2.2). Let us fix our set of *atoms* to be $\mathbb{A} = \mathbb{N} \setminus \{0\}$. Let $\mathcal{I}$ be a single-component 2D FO interpretation specifying a function $f\colon \Gamma^* \to \Sigma^*$. The interpretation canonically defines a function $f^{(\mathbb{A})}\colon (\Gamma \times \mathbb{A})^* \to (\Sigma \times \mathbb{A}^2)^*$ in the following way: reusing the notation O_u associated to $\mathcal{I}$ in Definition 3.1,

$$f^{(\mathbb{A})}\begin{pmatrix} u_1 & \dots & u_n \\ a_1 & \dots & a_n \end{pmatrix} = \begin{matrix} v_1 & \dots & v_m \\ a_{i_1}, a_{j_1} & \dots & a_{i_m}, a_{j_m} \end{matrix} \quad \text{where} \quad \begin{matrix} f(u) = v_1 \dots v_m \\ O_u = \{\langle i_1, j_1\rangle \leq \dots \leq \langle i_m, j_m\rangle\} \end{matrix}$$

(this is morally a form of "origin semantics" as in §2.1). For instance, for the interpretation of Example 3.4 defining $f_1\colon a^n \mapsto (a^{n-1}b)^{n-1}$, we would have

$$f_1(aaa) = aabaab \qquad f_1^{(\mathbb{A})}\begin{pmatrix} a & a & a \\ 3 & 1 & 2 \end{pmatrix} = \begin{matrix} a & a & b & a & a & b \\ 3,3 & 3,1 & 3,2 & 1,3 & 1,1 & 1,2 \end{matrix}$$

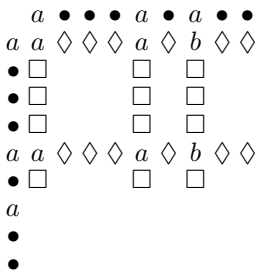
Next, choose $\bullet \notin \Gamma$ and $\square, \lozenge \notin \Sigma$ with $\square \neq \lozenge$. We define the codings

$$\mathtt{enc}_1\colon \begin{matrix} c \\ n \end{matrix} \in \Gamma \times \mathbb{A} \mapsto c\bullet^n \in (\Gamma \cup \{\bullet\})^* \qquad \mathtt{enc}_2\colon \begin{matrix} c' \\ p, q \end{matrix} \in \Sigma \times \mathbb{A}^2 \mapsto c'\square^p\lozenge^q \in (\Sigma \cup \{\square, \lozenge\})^*$$

which we extend to input strings as morphisms of free monoids.

▷ Claim 4.10. There is a function f' specified by some 2D FO interpretation $\Psi(\mathcal{I})$ such that $f' \circ \mathtt{enc}_1 = \mathtt{enc}_2 \circ f^{(\mathbb{A})}$ and $f'(\bullet w) = f'(w)$ for every input word w.

For example, $\mathcal{I}_2 = \Psi(\text{Example 3.4})$ can be illustrated (similarly to Example 3.5) by



$$\rightsquigarrow \quad a\square\square\square\lozenge\lozenge\lozenge a\square\square\square\lozenge b\square\square\square\lozenge\lozenge a\square\lozenge\lozenge\lozenge a\square\lozenge b\square\lozenge\lozenge = \mathtt{enc}_2 \circ f_1^{(\mathbb{A})}\begin{pmatrix} a & a & a \\ 3 & 1 & 2 \end{pmatrix}$$

Proof. Given a formula F over the relational signature for Γ^*, let F^{R} be the relativized formula over $(\Gamma \cup \{\bullet\})$ where all quantifiers $\forall z.(\dots)$ and $\exists z.(\dots)$ are replaced by $\forall z.\, \neg \bullet(z) \Rightarrow (\dots)$ and $\exists z.\, \neg \bullet(z) \wedge (\dots)$ respectively. Let

$$P(x, y) = y \leq x \wedge \neg \bullet(y) \wedge \forall z.\, (\neg(z \leq y) \wedge z \leq x) \Rightarrow \bullet(z)$$

We take the following definition for the interpretation $\Psi(\mathcal{I})$:

[19] This implies $(a^n \mapsto (a^{n-1}b)^{n-1}) \notin \mathsf{MTT}^1$, a fact that was later reproved in [40, Theorem 8.1(a)].

[20] The theorem says that some superclass of $\mathrm{Im}(\mathrm{SO}(\mathsf{MTT}^1))$ does not contain any language of the form $\{(a^n b)^{f(n)} \mid n \in X\}$ where $X \subseteq \mathbb{N} \setminus \{0\}$ is infinite and $f\colon X \to \mathbb{N} \setminus \{0\}$ is injective.

- $\Psi(\mathcal{I})_c(x_1,x_2) = \neg\bullet(x_1) \wedge \neg\bullet(x_2) \wedge \mathcal{I}_c^{\mathrm{R}}(x_1,x_2)$ for $c \in \Sigma$
- $\Psi(\mathcal{I})_\Box(x_1,x_2) = \bullet(x_1) \wedge \neg\bullet(x_2) \wedge \exists y.\, P(y,x_2) \wedge \bigvee_{c\in\Sigma} \mathcal{I}_c^{\mathrm{R}}(y,x_2)$
- $\Psi(\mathcal{I})_\Diamond(x_1,x_2) = \neg\bullet(x_1) \wedge \bullet(x_2) \wedge \exists y.\, P(x_1,y) \wedge \bigvee_{c\in\Sigma} \mathcal{I}_c^{\mathrm{R}}(x_1,y)$
- $\Psi(\mathcal{I})_\leq(x_1,x_2,y_1,y_2) = \exists \widehat{x}_1,\widehat{x}_2,\widehat{y}_1,\widehat{y}_2.\ \bigwedge_{i=1,2} P(x_i,\widehat{x}_i) \wedge P(y_i,\widehat{y}_i)$ and
  - either $\neg\mathcal{I}_{\leq}^{\mathrm{R}}(\widehat{y}_1,\widehat{y}_2,\widehat{x}_1,\widehat{x}_2)$
  - or $(\widehat{x}_1,\widehat{x}_2) = (\widehat{x}_2,\widehat{y}_2)$ and $(x_2,x_1) \leq (y_2,y_1)$ lexicographically. ◁

Let f' be given by Claim 4.10. We must now check that $\mathrm{Im}(f')$ is d-complete for $\mathrm{Im}(f)$.

- The part concerning the erasing morphism is immediate.
- So what remains is to verify δ-completeness (Definition 4.7). Let $u \in \mathrm{Im}(f)$, i.e. $u = f(v)$ for some $v \in \Gamma^*$, and $m = |v|$. Then $f'(v[1]\bullet v[2]\bullet\bullet\ldots v[m]\bullet^m) \in \mathrm{Im}(f')$ is of the required form. (It is here that we use the fact that our FO interpretation is single-component to ensure that the maximal factors of the output in $\Box^*\Diamond^*$ are pairwise distinct.) ◀

5 Polyregular word sequences

5.1 For-transducers in an abstract style

Tito — $\prod_{i=a}^{b}$ in non-commutative monoid

Recall the notations $\underline{\Sigma}$ and $w\not\downarrow i$ from the end of the introduction.

▶ **Definition 5.1.** *Let M be any monoid.*
Define the higher-order functions $\mathtt{for}^\uparrow, \mathtt{for}^\downarrow \colon [(\Sigma \cup \underline{\Sigma})^* \to M] \to [\Sigma^* \to M]$ *by*

$$\begin{array}{rccc} \mathtt{for}^\uparrow(f): & \Sigma^* & \to & M \\ & w & \mapsto & \prod_{i=1}^{|w|} f(w\not\downarrow i) \end{array} \qquad \begin{array}{rccc} \mathtt{for}^\downarrow(f): & \Sigma^* & \to & M \\ & w & \mapsto & \prod_{j=0}^{|w|-1} f(w\not\downarrow |w|-j) \end{array}$$

▶ **Proposition 5.2.** *The class of (string-to-string) polyregular functions can be characterized as the smallest class containing constant functions, closed under concatenation, regular conditionals and the* $\mathtt{for}^\uparrow$ *and* $\mathtt{for}^\downarrow$ *operators. For first-order polyregular functions, an analogous statement holds by replacing "regular conditional" by "star-free conditional".*

Proof idea. It is clear that all such functions are polyregular. The converse corresponds to converting the mutable finite state of a for-transducer [5, §1] into regular conditionals. ◀

Call the "product rank" of a polyregular function the minimum number of nestings of $\mathtt{for}^\uparrow$ and $\mathtt{for}^\downarrow$ operators necessary to define it.

▷ Claim 5.3 (looks obvious, but is it?). If f is polyregular and g is a concatenation of rational functions, then the product rank of $f \circ g$ is at most that of f.

Tito — prove this

An example of concatenation of rational functions is given, for any *fixed* $n \in \mathbb{N}$ and $\sigma \in \mathfrak{S}_n$, by $\widetilde{\sigma} \colon a^{x_1}\#\ldots\#a^{x_n} \mapsto a^{x_{\sigma(1)}}\#\ldots\#a^{x_{\sigma(n)}}$.

▶ **Definition 5.4.** *A function $f\colon \mathbb{N}^n \to \Sigma^*$ is polyregular if there exists a polyregular function $\hat{f}\colon \{a,\#\}^* \to \Sigma^*$ such that $f(x_1,\dots,x_n) = \hat{f}(a^{x_1}\#\dots\#a^{x_n})$ for all $x_1,\dots,x_n \in \mathbb{N}$. We then say that $\hat{h}$* represents *f (or* is a representative *of f).*

Our goal is to prove the following, which for $n = 1$ characterizes the polyregular functions with unary input.

▶ **Theorem 5.5.** *The polyregular functions $\mathbb{N}^n \to \Sigma^*$ for $n \geq 1$ can be characterized inductively as follows:*

- *Constant functions are polyregular.*
- *The concatenation of two polyregular functions (with the same arity) is polyregular.*
- *If $f\colon \mathbb{N}^n \to \Sigma^*$ is polyregular and $\sigma \in \mathfrak{S}_n$ then $f \circ \widetilde{\sigma}$ is polyregular.*
- *Indexed product: if $f\colon \mathbb{N}^{n+1} \to \Sigma^*$ is polyregular, then so is*

$$(x_1,\dots,x_n) \mapsto \prod_{y=0}^{x_1-1} f(y, x_1 - 1 - y, x_2, \dots, x_n)$$

- *Ultimately periodic combination: if g_i and h_j are polyregular, $k \geq 0$ and $p \geq 1$ then so is*

$$(x_1,\dots,x_n) \mapsto \begin{cases} g_{x_1}(x_2,\dots,x_n) & \text{when } x_1 < k \\ h_{(x_1-k) \bmod p}(\lfloor (x_1-k)/p \rfloor, x_2, \dots, x_n) & \text{when } x_1 \geq k \end{cases}$$

By restricting to $p = 1$ in the ultimately periodic combinations, we get a characterization of the FO-polyregular functions $\mathbb{N}^n \to \Sigma^*$.

Proof. The fact that all functions inductively generated this way are polyregular is routine. We therefore focus on the converse, by induction first on $r \in \mathbb{N}$ and then on Proposition 5.2 restricted to the functions with representatives of product rank $\leq r$. One key case is to handle representatives defined by regular conditionals, which is done by nesting n ultimately periodic combinations (one for each variable) thanks to an idempotent power argument. Another key case is as follows.

Let $f\colon \{a,\#,\underline{a},\underline{\#}\} \to \Sigma^*$ be of product rank $\leq r$, and let $\hat{g}$ be the function defined by the indexed product in Proposition 5.2. Let $g\colon \mathbb{N}^n \to \Sigma^*$ be represented by $\hat{g}$. We want to show that g fits our inductive definition.

Let $\chi_i(x_1,\dots,x_{i-1},y,z,x_{i+1},\dots,x_n) = a^{x_1}\#\dots\#a^{x_{i-1}}\#a^y\underline{a}a^z\#a^{x_{i+1}}\#\dots\#a^{x_n}$ and $\chi'_i(x_1,\dots,x_n) = a^{x_1}\#\dots\#a^{x_i}\underline{\#}a^{x_{i+1}}\#\dots\#a^{x_n}$. We then have, for $x_1,\dots,x_n \in \mathbb{N}$,

$$g(x_1,\dots,x_n) = \left(\prod_{i=1}^{n-1} u_i \cdot v_i\right) \cdot u_n$$

where for each i, we have $v_i = f \circ \chi'_i(x_1,\dots,x_n)$ and, assuming $x_i \geq 1$,

$$u_i = \prod_{y=0}^{x_i-1} f \circ \chi_i(\dots, x_{i-1}, y, x_i - 1 - y, x_{i+1}, \dots)$$

Since χ_i and χ'_i have rational representatives, according to Claim 5.3, $f \circ \chi_i$ and $f \circ \chi'_i$ are represented by polyregular functions with product rank $\leq r$, allowing us to invoke the induction hypothesis, and we are then done. (Well, almost, because we need to handle the edge case $x_i = 0$ and get rid of the -1 to fit with our actual inductive definition – but this can be done using an ultimately periodic combination.) ◀

5.2 Minimizing the number of loops for polyregular sequences

Warning: Very rough and sketchy draft; we ignore the permutations of variables throughout for convenience

Cécilia — ↑

Now we will turn to showing a pebble minimization theorem for polyregular word sequences. Theorem 5.5 gives us an inductive characterization of such sequence as being closed under a number of operators that include ultimately periodic combination. Now we shall show that those operators can always be brought up at the toplevel.

▶ **Definition 5.6.** *Call* simple polyregular word matrices $\mathbb{N}^n \to \Sigma^*$ *for* $n \geq 1$ *the smallest class of functions containing constant functions and closed under concatenation, permutation of input variables and indexed products.*

▶ **Theorem 5.7.** *Polyregular functions* $\mathbb{N}^n \to \Sigma^*$ *correspond exactly to ultimately periodic combinations of simple polyregular word matrices.*

Proof. The proof goes by an induction over the syntax corresponding to the inductive description given in Theorem 5.5, observing beforehand a couple of facts

Now to set up the induction, call a *polyregular AST* (abstract syntax tree) a syntactic representation of a polyregular function as per the description given in Theorem 5.5. Given an AST T, assign a multiset of natural numbers as follows:

- the AST corresponding to constant functions get assigned the empty multiset.
- given two ASTs T_1 and T_2, the multiset corresponding to the concatenation AST $T_1 \cdot T_2$ will be $\mu(T_1 \cdot T_2) = \mu(T_1)^\uparrow + \mu(T_2)^\uparrow$, where $X^\uparrow = \{i+1 \mid i \in X\}$.
- given an AST T of an indexed product, if T' is the AST corresponding to the inner component, the resulting AST will be $\mu(T) = \mu(T')^\uparrow$
- the AST corresponding to a ultimately periodic combination $\mathsf{upc}(x_1; T'_1, \dots, T'_{k-1}; T''_0, \dots, T''_{p-1})$ will have measure $\mu(T) = \max\{T'_i, T''_j \mid 0 < i < k, j \leq p\}$

We show that any AST associated to a multi-set containing a non-zero element can be rewritten to an AST associated to a strictly small multiset in the canonical well-founded order. Such rewrites are terminating and will give us the desired result. To do so, we need a few auxiliary lemmas.

▶ **Lemma 5.8.** *If* $f \colon \mathbb{N}^{n+1} \to \Sigma^*$ *is a simple numerical polyregular function, for any* $k \in \mathbb{N}$*, the function* $g \colon (x_2, \dots, x_n) \mapsto f(k, x_2, \dots, x_n)$ *is also a simple polyregular word sequence. Further, if* T_f *is an AST representing* f*, we can build an AST* T_g *representing* g *such that* $\mu(T_g) \leq \mu(T_f)$*.*

Proof. Turn indexed product involving the x_1 component into a finitary concatenation. ◀

▶ **Lemma 5.9.** *If* $f \colon \mathbb{N}^{n+1} \to \Sigma^*$ *is a simple numerical polyregular function, for any* $a, b \in \mathbb{N}$*, the function* $g \colon (x_1, x_2, \dots, x_n) \mapsto f(ax_1 + b, x_2, \dots, x_n)$ *is also a simple polyregular word sequence. Further, if* T_f *is an AST representing* f *with* $\mu(T_f)$ *containing only* 0*s, we can build an AST* T_g *representing* g *such that* $\mu(T_g) = \mu(T_f)$*.*

Proof. By induction over the input AST, unrolling a times each loop containing the variable x_1 as a bound and adding b initial executions. ◀

Now, let's carry on with the induction. Fix an AST and consider a critical maximally deep ultimately periodic combination therein; it is necessarily preceded by either a concatenation or an indexed product node.

- Suppose that we have a concatenation involving an ultimately periodic combination

$$\mathsf{upc}(x_1; f_0, f_1, \dots, f_{k-1}; g_0, \dots, g_{p-1}) \cdot h$$

where x_1 is the variable we are making the combination over, the f_i representing the functions corresponding to the offset and the g_j the functions corresponding to the period of the first component. For $i < k$, define h'_i so that the corresponding function is

$$(x_2, \dots, x_n) \mapsto h(i, x_2, \dots, x_n)$$

and h''_j for $i < p$ by

$$(x_1, x_2, \dots, x_n) \mapsto h(k + px_1, x_2, \dots, x_n)$$

By Lemma 5.8 and Lemma 5.9, those two functions can be represented by ASTs with smaller measure than h. We can then represent the original function with the AST

$$\mathsf{upc}(x_1; f_0 h'_0, \dots, f_{k-1} h'_{k-1}; g_0 \cdot h''_0, \dots, g_{p-1} \cdot h''_{p-1})$$

which has strictly smaller measure.

- The case where the ultimately periodic combination as the second component of a concatenation is handled in the same way.
- The key case is the one of the indexed product. There are two subcases
  - Either the combination operates according to a variable that is not iterated over in the indexed product; in such a case, the combination trivially commutes with the indexed product.
  - Otherwise, assume that the combination is according to the first variable created by the indexed product, i.e., that the function F under consideration is

$$(x_1, \dots, x_n) \mapsto \prod_{y=0}^{x_1 - 1} f(y, x_1 - 1 - y, \dots)$$

with $k \geq 0$, $p \geq 1$ such that $f(y, z, \dots) = g_y(z, \dots)$ for $y < k$ and $f(y, z, \dots) = h_{(y-k) \bmod p}(\lfloor (y-k)/p \rfloor, z, \dots)$ with g_y and the h_i simple. We express it as a new combination using offset $2k$ but keeping the period p using a sort of loop unrolling. The idea For $x_1 < k$, we can consider the functions l_{x_1}

$$l_{x_1} \colon (x_2, \dots, x_n) \mapsto \prod_{z=0}^{x_1 - 1} g_z(x_1 - 1 - z, \dots)$$

and for $k \geq x_1 < 2k$

$$l_{x_1} \colon (x_2, \dots, x_n) \mapsto \left(\prod_{z=0}^{k-1} g_z(x_1 - 1 - z, \dots) \right) \cdot \left(\prod_{z=k}^{x_1} h_{(z-k) \bmod p}(\lfloor (z-k)/p \rfloor, \dots, x_n) \right)$$

which can all be represented by ASTs of smaller measure thanks to our lemmas. For the period, we consider the functions m_j for $j < p$ defined as

$$m_j \colon (x_1, \dots, x_n) \mapsto l_{k+j}(x_2, \dots, x_n) \cdot \prod_{y=0}^{x_1 - 1} \theta_j(y, x_1 - 1 - y, \dots)$$

where

$$\theta_j \colon (y, z, x_2, \dots x_n) \mapsto \prod_{k=j}^{p+j-1} h_{k \bmod p}(y, \dots, x_n)$$

All of those also correspond to AST of smaller measure, so overall the ultimately periodic combination has strictly smaller measure than before.

- The case where the combination is according to the second variable created by the indexed product is handled in a similar way. ◀

Once we have converted a polyregular word sequence into a combination of simple polyregular matrices, the asymptotic growth is necessarily bounded by the growth of a dominant simple polyregular matrix in the decomposition. Further, it is easy to normalize an expression corresponding to a simple polyregular matrix by removing subexpressions containing only empty outputs so that the asymptotic growth of the associated function is $O(n^d)$ where d is the number of imbricated loops. To obtain the minimization theorem, it suffices to show such combinations of simple polyregular matrices can be turned into d-pebble transducers.

Cécilia — the paragraph above should be expanded somewhat.

Tito — mention that it entails pebble minimization in the usual sense

5.3 Polyregular integer sequences (unary output too)

We may identify polynomials with the functions they denote in the sequel: $A[X] \subset A^A$. (Recall that when A is an infinite integral domain, this identification is injective.)

▶ Remark 5.10 (Related work). The specialization of [40, Theorem 9.2] to unary outputs says that a sequence $\mathbb{N} \to \mathbb{N}$ is polyblind of rank at most $k \in \mathbb{N}$ if and only if it is an ultimately periodic combination of *polynomials* in $\mathbb{N}[X]$ of degree at most $k+1$. The regular case ($k = 0$) was first shown in [8].

This should be included in the *poly-rational sequences* $\mathbb{N} \to \mathbb{Q}$ introduced recently in [1] (counter-intuitively those don't have polynomial growth). Poly-rational sequences form a subclass of linear recursive sequences (LRS) over $\mathbb{Q}$ (see [1, Theorem 1] for several equivalent definitions), while the notion of "catenative recurrence" which generalizes LRS over $\mathbb{N}$ to non-unary inputs characterizes $\mathbb{N}$-rational series (i.e. HDT0L with unary ouptut) [29].

For bigger classes of automata-theoretic sequences $\mathbb{N} \to \mathbb{N}$, see [31, 7]; and for polyregular sequences with $\mathbb{Z}$-valued output see [9].

Non-aperiodic case We refer to Section 5 for the notion of ultimately periodic combination.

▶ **Theorem 5.11.** *Polyregular, polyblind and marble (Definition 5.13) integer sequences all correspond to ultimately periodic combination of integer polynomials.*

(This collapse of the three classes in the case $\mathbb{N} \to \mathbb{N}$ has been independently proved in [13, §4] as a corollary of a much stronger result.)

By [40, Theorem 9.2], we know that every sequence of $\mathsf{UPC}(\mathbb{N}[X])$ is polyblind, and a fortiori polyregular. By [12, Corollary 17], we know that polyregular and marble sequences coincide. So it suffices to show the following.

▶ **Lemma 5.12.** *Every marble sequence is in* $\mathsf{UPC}(\mathbb{N}[X])$.

To do so, let us employ the definition of marble sequence in terms of bimachines of [12]. Restricted to the setting of input over unary alphabets, the definition is equivalent to the following.

▶ **Definition 5.13.** *Let $S \subseteq \mathbb{N}^{\mathbb{N}}$. A marble S-bimachine (also called marble bimachine with external functions in S) is a tuple $\mathcal{M} = (M, x, \mathfrak{F}, \lambda)$ with*

- *M is a finite monoid and $x \in M$*
- *$\mathfrak{F}$ is a finite subset of S*
- *λ is a function $M \times M \to \mathfrak{F}$*

The output function associated with $\mathcal{M}$ (written abusively $n \mapsto \mathcal{M}(n)$) is defined as

$$\mathcal{M}(n) \;=\; \sum_{i=0}^{n} \lambda(x^i, x^{n-i})(i)$$

Define inductively the notion of n-marble bimachine and the associated set of integer sequences Marble_n:

- *A 0-marble bimachine is a bimachine with external functions which are constant.*
- *A $n+1$-marble bimachine is a marble Marble_n-bimachine.*

Lemma 5.12 thus states that $\mathsf{Marble}_n \subseteq \mathsf{UPC}(\mathbb{N}[X])$. This is proven by induction over n. The base case is trivial, and the inductive case can be reduced to the following lemma.

▶ **Lemma 5.14.** *Every $\mathsf{UPC}(\mathbb{N}[X])$-bimachine computes a function in $\mathsf{UPC}(\mathbb{N}[X])$.*

Proof. Let $\mathcal{M} = (M, x, \mathfrak{F}, \lambda)$ be the bimachine under consideration. Let $m \in \mathbb{N} \setminus \{0\}$ be such that $x^m = x^{2m}$ and, for every $f \in \mathfrak{F}$, there exist polynomials $(P_i)_{i=0}^{m-1}$ such that $f((n+1)m + k) = P_k(n)$ for every $k < m$.

To conclude, it suffices to show that the functions $g_k \colon n \mapsto \mathcal{M}((n+2)m + k)$ are in $\mathsf{UPC}(\mathbb{N}[X])$ for every $k < m$. We have

$$\begin{aligned} g_k(n) &= \sum_{i=0}^{(n+2)m+k} \lambda(x^i, x^{(n+2)m+k-i})(i) \\ &= \sum_{i=0}^{n+2} \sum_{j=0}^{m-1} \lambda(x^{im+j}, x^{(n+2)m+k-(im+j)})(im+j) + \sum_{i=(n+2)m}^{(n+2)m+k} \lambda(x^i, x^{(n+2)m+k-i})(i) \end{aligned}$$

Exploiting idempotency of x^m, we thus have

$$\begin{aligned} g_k(n) = \sum_{i=1}^{n} \sum_{j=0}^{m-1} \lambda(x^{m+j}, x^{2m+k-j})(im+j) \;+\; g_k(m-1) \\ + \sum_{j=0}^{m+k} \lambda(x^{m+j}, x^{m+k-j})((n+1)m+j) \end{aligned}$$

Define $Q_k(n)$ as $g_k(m-1) \;+\; \sum_{j=0}^{m+k} \lambda(x^{m+j}, x^{m+k-j})((n+1)m+j)$, so that we have

$$g_k(n) = Q_k(n) + \sum_{i=1}^{n} \sum_{j=0}^{m-1} \lambda(x^{m+j}, x^{2m+k-j})(im+j)$$

Recall that we chose m so that $\lambda(x,y)((n+1)m+k) = P_{x,y,k}(n)$ for some polynomial $P_{x,y,k} \in \mathbb{N}[X]$. Hence, we have that Q_k is an integer polynomial and that there are polynomials $P_{k,j} \in \mathbb{N}[X]$ for $j < m$ such that

$$g_k(n) = Q_k(n) + \sum_{i=1}^{n} \sum_{j=0}^{m-1} P_{k,j}(i)$$

Taking P_k to be the sum $\sum_{j=0}^{m-1} P_{k,j}$, we thus have $g_k(n) = Q_k(n) + \sum_{i=1}^{n} P_k(i)$.

Now we can conclude that $g_k \in \mathsf{UPC}(\mathbb{N}[X])$ using the two following facts. The first follows from Faulhaber's formula for sums of powers.

▷ Claim 5.15. For every polynomial $P \in \mathbb{N}[X]$, there is $Q \in \mathbb{Q}[X]$ such that $\sum_{i=0}^{n} P(i) = Q(n)$.

The second is easy to check.

▷ Claim 5.16. $\{f : \mathbb{N} \to \mathbb{N} \mid \exists Q \in \mathbb{Q}[X].\forall n \in \mathbb{N}.\ f(n) = Q(n)\} \subseteq \mathsf{UPC}(\mathbb{N}[X])$

Proof. We can decompose the proof in two steps. First we show that polynomials $Q \in \mathbb{Q}[X]$ that preserve positive integers are periodic combination of polynomials $P \in \mathbb{Z}[X]$ that preserve integers, and then than any such integer polynomial composed with $X + N$ for a sufficiently large N yields a polynomial of $\mathbb{N}[X]$. We can then conclude using the fact that UPC is a closure operator.

First, assume that $Q \in \mathbb{Q}[X]$ with $Q(n) \in \mathbb{N}$ for every $n \in \mathbb{N}$. Then, take the least common multiple of the denominators of the coefficients of Q and call it m. For $k < m$, consider $Q_k(X) = Q(mX + k)$. The non-constant coefficients can be seen to be integers by expanding the expressions. The constant coefficient of Q_k is equal to $Q(k) \in \mathbb{N}$. Clearly, all the Q_k preserve positive integers, so we may conclude that $Q \in \mathsf{UPC}(\{f \mid \exists P \in \mathbb{Z}[X].\forall n \in \mathbb{N}.\ f(n) = P(n)\})$.

Now suppose that $P \in \mathbb{Z}[X] \setminus \{0\}$ and that it preserves positive integers (the case $P = 0$ is trivial as $0 \in \mathbb{N}[X]$). Let d be the degree of P and $P(X) = \sum_{i=0}^{d} p_i X^i$. Take $N = d! \max_{i \le d} |p_i|$ and consider the polynomial $R(X) = P(X + N)$. R and P have the same leading coefficient p_d, which is necessarily positive because $\lim_{n \to +\infty} P(n) > 0$. The kth coefficient of R is $\sum_{i=k}^{d} \binom{i}{k} N^{i-k} p_i$, which is positive because

$$\begin{aligned}\binom{d}{k} N^{d-k} p_d \ge N^{d-k} \ge N^{d-k}(d-1-k)\binom{d-1}{k} \max_{i \le d} |p_i| \ge N^{d-k-1} \max_{i \le d} |p_i| \sum_{i=k}^{d-1} \binom{i}{k} \\ \ge \sum_{i=k}^{d-1} \binom{i}{k} N^{i-k} p_i\end{aligned}$$

Whence $P \in \mathsf{UPC}(\mathbb{N}[X])$. ◁

By combining the g_k for $k < m$, we have $\mathcal{M} \in \mathsf{UPC}(\mathsf{UPC}(\mathbb{N}[X])) = \mathsf{UPC}(\mathbb{N}[X])$. ◀

First-order case

▶ **Theorem 5.17.** *A function $f : \mathbb{N} \to \mathbb{N}$ is FO polyblind of rank $\le k$ if and only if there exists $P \in \mathbb{Z}[X]$ of degree $\le k + 1$ such that for all $n \in \mathbb{N}$ except finitely many, we have $f(n) = P(n)$.*

Proof. By induction on the rank. ◀

▶ **Proposition 5.18** (useful for the "if" direction above). *Let $P \in \mathbb{Z}[X]$ and assume that its leading coefficient is positive. Then $P(X+n) \in \mathbb{N}[X]$ for all $n \in \mathbb{N}$ except finitely many.*

Proof. Write the Taylor formula for $P(X+n)$; the k-th derivatives of P for $k \leq \deg(P)$ have a positive leading coefficient and therefore take positive values for large enough n. ◀

▶ **Corollary 5.19** (A surprising phenomenon). *There exists a function $\mathbb{N} \to \mathbb{N}$ which is both polyblind and first-order polyregular, but not FO polyblind.*

Proof. Take $n \mapsto n(n+1)/2$. ◀

▶ **Theorem 5.20.** *For unary ouptuts, FO-marble = FO-polyregular.*

Proof. A sum of n FO queries can be translated into the sum of $n \times k!$ FO k-marble bimachines, each computing the part of the output produced when the variables used in the query are in a certain ordering relative to one another. (This is a different construction from the one used in [12] for the non-FO case which uses streaming string transducers. Note also that it means converting an logical interpretation to a pebble transducer is trivial in the unary output case.) ◀

▶ **Theorem 5.21.** *A function $f\colon \mathbb{N} \to \mathbb{N}$ is FO-polyregular if and only if there exists $P \in \mathbb{Q}[X]$ such that for almost all $n \in \mathbb{N}$, we have $f(n) = P(n)$.*

Proof. Note that this forces P to be integer-valued.

"if" Use Newton's interpolation formula

$$P(X+a) = \sum_{i=0}^{\deg P} \Delta^k(P)(a)\binom{X}{k} \quad \text{where} \quad \Delta(P) = P(X+1) - P(X)$$

and note that just like the k-th derivative, $\Delta^k(P)$ also takes positive values for large enough arguments. Computing binomial coefficients with FO-marble transducers is trivial.

"only if" Similar argument to Lemma 5.12, using the previous theorem. ◀

References

1 Corentin Barloy, Nathanaël Fijalkow, Nathan Lhote, and Filip Mazowiecki. A robust class of linear recurrence sequences. *Information and Computation*, 2022. doi:10.1016/j.ic.2022.104964.

2 Mikołaj Bojańczyk. Transducers with origin information. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 26–37. Springer, 2014. doi:10.1007/978-3-662-43951-7_3.

3 Mikołaj Bojańczyk. Polyregular functions, 2018. arXiv:1810.08760.

4 Mikołaj Bojańczyk. On the growth rate of polyregular functions, 2022. arXiv:2212.11631.

5 Mikołaj Bojańczyk. Transducers of polynomial growth. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 1:1–1:27. ACM, 2022. doi:10.1145/3531130.3533326.

6 Mikołaj Bojańczyk, Sandra Kiefer, and Nathan Lhote. String-to-string interpretations with polynomial-size output. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 106:1–106:14, 2019. doi:10.4230/LIPIcs.ICALP.2019.106.

7 Michaël Cadilhac, Filip Mazowiecki, Charles Paperman, Michał Pilipczuk, and Géraud Sénizergues. On polynomial recursive sequences. *Theory of Computing Systems*, June 2021. doi:10.1007/s00224-021-10046-9.

8 Christian Choffrut and Bruno Guillon. An algebraic characterization of unary two-way transducers. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, volume 8634 of *Lecture Notes in Computer Science*, pages 196–207. Springer, 2014. doi:10.1007/978-3-662-44522-8_17.

9 Thomas Colcombet, Gaëtan Douéneau-Tabot, and Aliaume Lopez. Z-polyregular functions, 2022. arXiv:2207.07450.

10 Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic. A language-theoretic approach.* Encyclopedia of Mathematics and its applications, Vol. 138. Cambridge University Press, June 2012. Collection Encyclopedia of Mathematics and Applications, Vol. 138. URL: https://hal.archives-ouvertes.fr/hal-00646514.

11 Patrice Ossona de Mendez. First-order transductions of graphs (invited talk). In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 2:1–2:7. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.2.

12 Gaëtan Douéneau-Tabot. Pebble Transducers with Unary Output. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*, volume 202 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2021.40.

13 Gaëtan Douéneau-Tabot. Hiding Pebbles When the Output Alphabet Is Unary. In Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 120:1–120:17, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2022.120.

14 Gaëtan Douéneau-Tabot. Pebble minimization: the last theorems, 2022. arXiv:2210.02426.

15 Gaëtan Douéneau-Tabot, Emmanuel Filiot, and Paul Gastin. Register Transducers Are Marble Transducers. In Javier Esparza and Daniel Kráľ, editors, *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2020.29.

16 Roger W. Ehrich and Stephen S. Yau. Two-way sequential transductions and stack automata. *Information and Control*, 18(5):404–446, 1971. doi:10.1016/S0019-9958(71)90483-9.

17 Joost Engelfriet. Three hierarchies of transducers. *Mathematical Systems Theory*, 15(2):95–125, 1982. doi:10.1007/BF01786975.

18 Joost Engelfriet. Two-way pebble transducers for partial functions and their composition. *Acta Informatica*, 52(7-8):559–571, 2015. doi:10.1007/s00236-015-0224-3.

19 Joost Engelfriet and Linda Heyker. The string generating power of context-free hypergraph grammars. *Journal of Computer and System Sciences*, 43(2):328–360, 1991. doi:10.1016/0022-0000(91)90018-Z.

20 Joost Engelfriet and Hendrik Jan Hoogeboom. MSO definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic*, 2(2):216–254, April 2001. doi:10.1145/371316.371512.

915 **21** Joost Engelfriet, Hendrik Jan Hoogeboom, and Bart Samwel. XML navigation and transformation by tree-walking automata and transducers with visible and invisible pebbles. *Theoretical Computer Science*, 850:40–97, January 2021. doi:10.1016/j.tcs.2020.10.030.

918 **22** Joost Engelfriet, Kazuhiro Inaba, and Sebastian Maneth. Linear-bounded composition of tree-walking tree transducers: linear size increase and complexity. *Acta Informatica*, 58(1-2):95–152, 2021. doi:10.1007/s00236-019-00360-8.

921 **23** Joost Engelfriet and Sebastian Maneth. Output string languages of compositions of deterministic macro tree transducers. *Journal of Computer and System Sciences*, 64(2):350–395, 2002. doi:10.1006/jcss.2001.1816.

924 **24** Joost Engelfriet and Sebastian Maneth. Two-way finite state transducers with nested pebbles. In Krzysztof Diks and Wojciech Rytter, editors, *Mathematical Foundations of Computer Science 2002, 27th International Symposium, MFCS 2002, Warsaw, Poland, August 26-30, 2002, Proceedings*, volume 2420 of *Lecture Notes in Computer Science*, pages 234–244. Springer, 2002. doi:10.1007/3-540-45687-2_19.

929 **25** Joost Engelfriet and Sebastian Maneth. A comparison of pebble tree transducers with macro tree transducers. *Acta Informatica*, 39(9):613–698, 2003. doi:10.1007/s00236-003-0120-0.

931 **26** Joost Engelfriet and Heiko Vogler. Macro tree transducers. *Journal of Computer and System Sciences*, 31(1):71–146, 1985. doi:10.1016/0022-0000(85)90066-2.

933 **27** Joost Engelfriet and Heiko Vogler. Pushdown machines for the macro tree transducer. *Theoretical Computer Science*, 42:251–368, 1986. doi:10.1016/0304-3975(86)90052-6.

935 **28** Joost Engelfriet and Heiko Vogler. High level tree transducers and iterated pushdown tree transducers. *Acta Informatica*, 26(1/2):131–192, 1988. doi:10.1007/BF02915449.

937 **29** Julien Ferté, Nathalie Marin, and Géraud Sénizergues. Word-Mappings of Level 2. *Theory of Computing Systems*, 54(1):111–148, January 2014. doi:10.1007/s00224-013-9489-5.

939 **30** Emmanuel Filiot and Pierre-Alain Reynier. Copyful streaming string transducers. *Fundamenta Informaticae*, 178(1-2):59–76, January 2021. doi:10.3233/FI-2021-1998.

941 **31** Séverine Fratani and Géraud Sénizergues. Iterated pushdown automata and sequences of rational numbers. *Annals of Pure and Applied Logic*, 141(3):363–411, September 2006. doi:10.1016/j.apal.2005.12.004.

944 **32** Olivier Gauwin. *Transductions: resources and characterizations*. Habilitation à diriger des recherches, Université de Bordeaux, October 2020. URL: https://tel.archives-ouvertes.fr/tel-03118919.

947 **33** Liam Jordon. *An Investigation of Feasible Logical Depth and Complexity Measures via Automata and Compression Algorithms*. PhD thesis, National University of Ireland Maynooth, 2022. URL: https://mural.maynoothuniversity.ie/16566/.

950 **34** Makoto Kanazawa, Gregory M. Kobele, Jens Michaelis, Sylvain Salvati, and Ryo Yoshinaka. The failure of the strong pumping lemma for multiple context-free languages. *Theory of Computing Systems*, 55(1):250–278, 2014. doi:10.1007/s00224-014-9534-z.

953 **35** Stephan Kreutzer and Cristian Riveros. Quantitative monadic second-order logic. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 113–122. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.16.

957 **36** Michel Latteux. Substitutions dans les EDT0L systèmes ultralinéaires. *Information and Control*, 42(2):194–260, 1979. doi:10.1016/S0019-9958(79)90641-7.

959 **37** Sebastian Maneth. A survey on decidable equivalence problems for tree transducers. *International Journal of Foundations of Computer Science*, 26(8):1069–1100, 2015. doi:10.1142/S0129054115400134.

962 **38** Tova Milo, Dan Suciu, and Victor Vianu. Typechecking for XML transformers. *Journal of Computer and System Sciences*, 66(1):66–97, 2003. Journal version of a PODS 2000 paper. doi:10.1016/S0022-0000(02)00030-2.

965 **39** Anca Muscholl and Gabriele Puppis. The Many Facets of String Transducers. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical*

Aspects of Computer Science (STACS 2019), volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. doi:10.4230/LIPIcs.STACS.2019.2.

40 Lê Thành Dũng Nguyễn, Camille Noûs, and Cécilia Pradic. Comparison-Free Polyregular Functions. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 139:1–139:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.139.

41 Erik Paul. On finite and polynomial ambiguity of weighted tree automata. In Srecko Brlek and Christophe Reutenauer, editors, *Developments in Language Theory - 20th International Conference, DLT 2016, Montréal, Canada, July 25-28, 2016, Proceedings*, volume 9840 of *Lecture Notes in Computer Science*, pages 368–379. Springer, 2016. Full proofs available in the diploma thesis https://www.informatik.uni-leipzig.de/~epaul/Diplom_Thesis.pdf titled *Weighted Tree Automata and Quantitative Logics with a Focus on Ambiguity*. doi:10.1007/978-3-662-53132-7_30.

42 Václav Rajlich. Absolutely parallel grammars and two-way finite state transducers. *Journal of Computer and System Sciences*, 6(4):324–342, 1972. doi:10.1016/S0022-0000(72)80025-4.

43 Jonathan Rawski, Hossep Dolatian, Jeffrey Heinz, and Eric Raimy. Regular and polyregular theories of reduplication. *Glossa: a journal of general linguistics*, 8(1), 2023. doi:10.16995/glossa.8885.

44 Brigitte Rozoy. Outils et résultats pour les transducteurs boustrophédons. *RAIRO – Theoretical Informatics and Applications*, 20(3):221–249, 1986. doi:10.1051/ita/1986200302211.

45 Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009. Translated by Reuben Thomas. doi:10.1017/CBO9781139195218.

46 Helmut Seidl, Sebastian Maneth, and Gregor Kemper. Equivalence of deterministic top-down tree-to-string transducers is decidable. *Journal of the ACM*, 65(4):21:1–21:30, 2018. doi:10.1145/3182653.

47 Tim Smith. A pumping lemma for two-way finite transducers. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, volume 8634 of *Lecture Notes in Computer Science*, pages 523–534. Springer, 2014. doi:10.1007/978-3-662-44522-8_44.

48 Howard Straubing. First-order logic and aperiodic languages: a revisionist history. *ACM SIGLOG News*, 5(3):4–20, 2018. doi:10.1145/3242953.3242956.

49 Géraud Sénizergues. Word-mappings of level 3, 2023. arXiv:2301.09966.