Comparing the power of advice strings: a notion of complexity for infinite words

Gaëtan Douéneau-Tabot*

gaetan.doueneau@ens-paris-saclay.fr

January 30, 2018

Abstract

We investigate in this paper a certain notion of comparison between infinite words. In a general way, if $M$ is a model of computation (e.g. Turing machines) and $C$ a class of objects (e.g. languages), the complexity of an infinite word $\alpha$ can be measured with respect to the objects from $C$ that are presentable with machines from $M$ using $\alpha$ as an oracle.

In our case, the model $M$ is finite automata and the objects are either recognized languages or presentable structures, known respectively as advice regular languages and advice automatic structures. This leads to several different classifications of infinite words that are studied in detail; logical and computational characterizations are derived. Our main results explore the connections between classes of advice automatic structures, MSO-transductions and two-way transducers. They suggest a closer study of the resulting hierarchy over infinite words.

1 Introduction

Several measures have been defined to describe the complexity of infinite strings. Among others we mention subword complexity [AS03], Kolmogorov complexity, and Turing degrees [Sac63]. Whereas the two first methods focus on the intrinsic information contained in a string, the other one studies the relation of computability from one word to another, defining a preorder whose properties are now quite well understood. Equivalently, this preorder compares the expressive power of Turing machines that use an infinite word as oracle.

This paper follows a similar idea: we consider finite automata that can access an infinite advice string while processing their input. Such automata define classes of advice regular languages [Sal68], that generalize standard regularity. This notion enables us to introduce a comparison for infinite words: $\alpha$ is simpler (in the sense of languages) than $\beta$ if every language recognized by an automaton with advice $\alpha$ can also be recognized with advice $\beta$, what corresponds to the intuition that $\alpha$ contains less information than $\beta$.

Before going further, we evoke the current motivations around advice regular languages. Standard regular languages can be used to encode finite signature structures, known as automatic structures. This concept, derived from Büchi’s

*École Normale Supérieure Paris-Saclay, Université Paris-Saclay, France
early automata-logic techniques, has been shown especially relevant since its formalization in \cite{KN95} and \cite{BG00}. The model opened the door to a vast range of decision procedures via automata constructions, but it suffers from a lack of expressiveness, since e.g. \( \langle \mathbb{Q}, + \rangle \) is not automatic \cite{Tsai11}. However, \( \langle \mathbb{Q}, + \rangle \) is an example of advice automatic structure: it can be encoded using advice regular languages (instead of standard regular languages) \cite{KRSZ12}. Such structures share many properties with the former automatic structures, furthermore the use of advices builds a rich framework to provide algorithmic meta-theorems \cite{AGR17}. We shall not follow a model-theoretic point of view on advice automatic structures, but we use them to define another notion of comparison over infinite words as follows: \( \alpha \) is simpler (in the sense of structures) than \( \beta \) if every automatic structure with advice \( \alpha \) is also automatic with advice \( \beta \).

This paper is structured as a quest for a relevant comparison model over infinite strings, via the notion of advice. The informal criteria we use to define a good complexity measure are the following: it should have a simple definition, be robust enough, but not too coarse because we want to separate understandable sequences. Note that Turing degrees do not match this intuition since they make no distinction between computable - useful - sequences.

Our results will establish an interesting correspondence between the expressive power of advices (compared more or less using languages) and certain forms of transductions, when considering the way they classify infinite strings. This is somehow surprising, since the theory of transformations between words tends to be more fruitful and more difficult than the study of languages, following an early remark of Dana Scott \cite{Sco67}: “the functions computed by the various machines are more important - or at least more basic - than the sets accepted by these devices”. The concept of advice helps unifying these two frameworks. Furthermore, we shall use this idea to provide slightly new perspectives on (advice) automatic presentations and logic over infinite words.

After recalling preliminary results on structures, formal languages and logic, we present formally in Section 3 the notion of regularity with advice, under several variants. We study the comparisons of words provided by classes of advice regular languages, as evoked above. An easy correspondence is drawn with transductions, for instance we show that every regular language with advice \( \alpha \) is also regular with \( \beta \) if and only if \( \alpha \) is the image of \( \beta \) under a Mealy machine. Nevertheless, we note that such comparisons are far from being robust.

Next we turn in Section 4 to classes of advice automatic structures and briefly study some standard properties. We then show that variants of advice regular languages in fact present the same classes of structures. This is a first step towards robustness of the notions. The proof of this result also provides an original normal form for MSO-formulas when interpreted in a fixed word model.

Section 5 intends to understand the comparison over infinite words defined with respect to advice automatic presentations (see above): it develops the most involved results of this paper. Similar investigations were built in \cite{LC07}, under the formalism of set-interpretations - a very close notion. We particularize their results to show that every automatic structure with advice \( \alpha \) is also automatic with \( \beta \) if and only if \( \alpha \) is the image of \( \beta \) under an MSO-transduction (some logical transformation between words). We then give a more handy equivalent statement: \( \alpha \) is the image of \( \beta \) under a two-way transducer. This result is quite specific and original, since such transducers are however not powerful enough to realize all functions of infinite words defined by MSO-transductions \cite{AFT12}.
We end this paper investigating more precisely the structure of this last preorder (defined in particular by two-way transductions) in Section 4. Even if no previous research was done on the subject, a very similar study was carried out in [EKSW15] for comparison via one-way finite transducers. In the light of their results, we rough out the structure of a new hierarchy and explain why a more involved questioning may be fruitful.

2 Preliminaries

2.1 Words and languages

Greek capitals Σ, Γ and Δ are used to denote alphabets, i.e. finite sets of letters. A (resp. infinite) word is an element of the free monoid Σ∗ (resp. an infinite sequence from Σω). Let w be a (possibly infinite) word, we denote its length by |w| ∈ N ∪ {ω}. If n ≥ 0 is a position, w[n] is the (n + 1)-th letter of w (when defined). If 0 ≤ m < n, let w[m : n] = w[m]w[m + 1]···w[n − 1] (when defined) and let w[m : m] be the empty word ε. We replace w[0 : n] by w[n] to denote a prefix, and write w[n] for the (possibly infinite) suffix w[n]w[n + 1]···.

Let □ be a symbol that never appears in the alphabets. We define an encoding for a pair of words, writing the same positions the one below the other and padding the possible holes with □.

Definition 2.1 (Convolution). If u and v are (possibly infinite) words, their convolution u ⊗ v is the word of length max(|u|, |v|) such that:

- (u ⊗ v)[n] = (u[n], v[n]) if n < min(|u|, |v|);
- (u ⊗ v)[n] = (u[n], □) if |v| ≤ n < |u|;
- (u ⊗ v)[n] = (□, v[n]) if |u| ≤ n < |v|.

Convolution is defined in a similar way for k-tuples of (finite or infinite) words. We assume familiarity with the standard results of automata theory. Denote by Reg (resp. ωReg) the class of regular (resp. ω-regular) languages.

We make a large use of logic-automata connections, especially over infinite words (see e.g. [Tho97] for a good survey).

2.2 Structures and logic

We shall deal with structures over a finite signature, denoted by fraktur letters A, B . . . When needed, structures are seen as purely relational (we replace the functions by their graphs). We associate to each infinite word α its word structure 2ωω = ⟨N, <, (Pα)α∈Γ⟩ where < is the usual ordering on positive integers, and n ∈ Pα if and only if α[n] = a. For succinctness reasons, α |= ϕ often stands for 2ωω |= ϕ. If τ is a signature and L a logic, L[τ]-formulas are L-formulas over the signature τ. We assume that equality implicitly belongs to every signature and write MSO[<, Γ] for MSO[<, (Pα)α∈Γ]. MSO-formulas can be interpreted using weak semantic (WMSO), where we allow set quantifications to range only over finite sets.

We write x ∈ C where C is a class of objects (and not necessarily a set) to mean that there is in C an object isomorphic to x. Structure isomorphisms are
bijections between the domains preserving the relations, language isomorphisms are re-labelling of the symbols. We use a similar definition for $\subseteq$.

The following definition formalises the idea of presenting structures with languages: we encode the elements of the domain as words, so that the relations can be described in consistent way.

**Definition 2.2.** Let $\mathfrak{A} := \langle A, R_1 \ldots R_n \rangle$ a relational structure and $\mathcal{C}$ a class of languages (possibly over infinite words). A $\mathcal{C}$-presentation of $\mathfrak{A}$ is a tuple $(L, L_1 \ldots L_n)$ of languages from $\mathcal{C}$ such that there exists a surjective function $\nu : L \rightarrow A$ with:

- $L_w = \{ w \otimes w' \mid w, w' \in L \text{ and } \nu(w) = \nu(w') \}$;
- for each $R_i$ of arity $r_i$, $L_i = \{ w_1 \otimes \cdots \otimes w_{r_i} \mid \forall 1 \leq j \leq r_i, w_j \in L \text{ and } (\nu(w_1), \ldots, \nu(w_{r_i})) \in R_i \}$.

The function $\nu$ describes how $A$ is encoded in $L$. Since we never deal with the elements directly, it does not belong explicitly to the presentation and can be considered as a notation. The alphabet of $L$ is called encoding alphabet and often denoted $\Sigma$. The presentation is said injective if $L_w := \{ w \otimes w \mid w \in L \}$.

The point is to find a class of languages robust and decidable enough. If $\mathcal{C}$ is the class of recursive languages, the $\mathcal{C}$-presentable structures correspond to early studied recursive structures [Mi].

More recently, the class of $(\omega)$Reg-presentable structures generated much attention, under the name of $(\omega)$-automatic structures [BG00]. Such structures can be described using a tuple of automata for the languages of the presentations. We denote by $(\omega)$AutStr the class of $(\omega)$-automatic structures.

**Example 2.3.** $\langle \mathbb{N}, +, 0, 1 \rangle \in$ AutStr.

Automata theory brings literally hundreds of nice properties in this field, the most famous (and useful in practise) being probably the following.

**Proposition 2.4** ([BG00]). Every $(\omega)$-automatic structure has a decidable $\mathsf{FO}$-theory. The method is effective when we have an explicit presentation.

One of the main current challenges is to describe which structures have an automatic presentation, and elegant characterizations have been stated for certain classes, such as finitely generated groups [OT05]. However, as shown in Theorem 2.5 the presentation fails for simple structures with decidable $\mathsf{FO}$-theory. This motivates the study of possible extensions.

**Theorem 2.5** ([Tsa11]). $\langle \mathbb{Q}, + \rangle$ is not an $(\omega)$-automatic structure.

### 2.3 Interpretations

A useful tool in model theory is the concept of interpretation, describing a structure in another (host) structure via a tuple of logical formulas.

**Definition 2.6** (interpretation). Let $\mathfrak{A}$ be a structure over a signature $\tau$, $\mathcal{L}$ be a logic and $\mathcal{L} := (\phi_0(x), \phi_1(x, y), \phi_1(x_1, \ldots, x_{r_1}), \ldots, \phi_p(x_1, \ldots, x_{r_p}))$ a tuple of $\mathcal{L}[\tau]$-formulas where $\overline{x}, \overline{y}$ and the $\overline{x}_i$ are $k$-tuples of free variables. Let

- $A_{\delta} := \{ \overline{a} = (a_1 \ldots a_k) \mid \mathfrak{A} \models \phi_\delta(\overline{a}) \};$
• a binary relation on $A_δ$ with $a \sim b$ if and only if $A_δ(\bar{a}, \bar{b})$;

• for $1 \leq i \leq p$, $R_i$ is a relation on $A_δ$ defined as $(\bar{a}_1 \ldots \bar{a}_r) \in R_i$ if and only if $A_δ(\bar{a}_1 \ldots \bar{a}_r) = i$.

We say that $I$ is a $k$-dimensional $L$-interpretation of a structure $B$ in the structure $A$ if the following conditions are met:

• $\sim$ defines an congruence relation on $A_δ$ with respect to $R_1 \ldots R_p$;

• $\langle A_δ, R_1 \ldots R_p \rangle / \sim$ is isomorphic to $B$.

The interpretation is said injective if $\sim$ is the equality relation of $A_δ$. In the literature, interpretations are often directly assumed to be 1-dimensional injective interpretations. The choice of the logic $L$ gives several kinds of interpretation, detailed in Definition 2.7.

**Definition 2.7.**

(i) An FO-interpretation is a tuple of FO-formulas. The elements of $A$ are encoded as tuples of elements in the host structure $B$.

(ii) An MSO-interpretation is a tuple of MSO-formulas with free first-order variables. If we use the weak semantic, we speak of WMSO-interpretation. Once more, the elements of $A$ are encoded as tuples of elements of $B$.

(iii) An $S$-interpretation (set) is a tuple of MSO-formulas with free set variables. If we use weak semantic, we speak of FS-interpretation (finite set). The elements of $A$ are encoded as tuples of (finite) sets of elements in the host structure.

We briefly discuss the behavior of interpretations with respect to composition. The presence of sets and the use of several dimensions forces to be careful in the statements of Fact 2.8 as discussed in Remark 2.9.

**Fact 2.8 (closure under composition).**

(i) If $A$ is FO-interpretable in $B$ which is FO-interpretable in $C$, then $A$ is directly FO-interpretable in $C$.

(ii) If $A$ is MSO-interpretable in $B$ which is 1-dimensionally MSO-interpretable in $C$, then $A$ is directly MSO-interpretable in $C$.

**Proof idea (folklore).** The formulas of the interpretation in $B$ can equivalently be described in $A$, up to adding some variables if necessary.

**Remark 2.9.**

(i) If $A$ is (1-dim.) $S$-interpretable in $B$ which is (1-dim.) $S$-interpretable in $C$, there is no reason why $A$ should be $S$-interpretable in $C$. Indeed, the elements of $C$ (i.e. sets of $B$) should be coded as sets of sets of $A$.

(ii) The case of MSO is a bit more subtle. If $A$ is MSO-interpretable in $B$ which is $k$-dim. MSO-interpretable in $C$ with $k > 1$, there is no reason why $A$ should directly be MSO-interpretable in $C$. Indeed, the sets of $B$ (that can be used in the last interpretation) are sets of $k$-tuples of elements of $C$, but we can only describe $k$-tuples of sets, what is formally different.
Remark 2.10. The composition properties allow - in specific cases - to transfer the decidability of the logical theory from the host structure to the other.

The notion of interpretation is a key concept to extend standard automata-logic equivalences to automatic structures.

Proposition 2.11 (KNRS04). A structure $\mathcal{A}$ is automatic (resp. $\omega$-automatic) if and only if $\mathcal{A}$ is $F\mathcal{S}$-interpretable (resp. $S$-interpretable) in $(\mathbb{N}, \prec)$.

3 Simple case: regular languages with advice

We introduce in this section an extension of regular languages known as regular languages with advice. This concept enables us to study some preorders over infinite words; we discuss their relevance and establish a first link with transductions. The fruits we catch here are hanging close to the ground, but they raise intuitions about the climbing that follows.

3.1 Terminating languages

The idea of advice regularity is to consider languages accepted by automata whose transitions changes in function of the position in the input word [Sal68]. Equivalently, we can consider that such a modified automaton reads an infinite advice string while processing its input [BS69]. Definition 3.1 describes a similar idea, but without dealing directly with automata.

Definition 3.1. $L \subseteq \Sigma^*$ is terminating regular with advice $\alpha \in \Gamma^\omega$ if there exists a regular language $L' \subseteq (\Sigma \times \Gamma)^*$ such that $L = \{ w \mid w \otimes \alpha_n \mid |w| \in L' \}$.

Example 3.2.

(i) If $L \subseteq \Sigma^*$ is regular, so is $\{ w \otimes w' \mid w \in L, w' \in \Gamma^*, |w| = |w'| \}$, and considering this language shows that $L$ is regular with any advice;

(ii) the set $\text{Pref}(\alpha) := \{ \alpha_n \mid n \geq 0 \}$ is regular with advice $\alpha$.

Remark 3.3. There are non-computable languages regular with some advice. We denote by $\text{Reg}[\alpha]$ be the class of regular languages with advice $\alpha$ (over any alphabet). As evoked in the introduction, our goal is to measure the complexity of infinite words, through the expressiveness of their advice classes. We shall say that $\alpha$ is simpler than $\beta$ if $\text{Reg}[\alpha] \subseteq \text{Reg}[\beta]$. Following this intuition, ultimately periodic words (i.e. infinite words of the form $w\omega^*$) are the simplest advices, as detailed in the next fact.

Fact 3.4. [BS69, Rei13] Example 3.2 provides $\forall \alpha, \text{Reg} \subseteq \text{Reg}[\alpha]$. If $\alpha$ is ultimately periodic, the information it contains can be hardcoded in a regular language, thus $\text{Reg} = \text{Reg}[\alpha]$. If $\alpha$ is not ultimately periodic, it can be shown that $\text{Pref}(\alpha) \notin \text{Reg}$, thus $\text{Reg} \not\subseteq \text{Reg}[\alpha]$.

We write $\alpha \preceq_{\text{Reg}} \beta$ when $\text{Reg}[\alpha] \subseteq \text{Reg}[\beta]$, this relation is a preorder on the class of infinite words. Let the $\preceq_{\text{Reg}}$-degrees be the equivalence classes of the relation $\preceq_{\text{Reg}} \cap \succ_{\text{Reg}}$, they describe the sets of equally complex advices. Note that ultimately periodic words form the least $\preceq_{\text{Reg}}$-degree.
One goal of this paper is to build generic equivalences between the power of advices and certain forms of transductions. The first relationship will be done between \( \preceq_{\text{Reg}} \) and Mealy machines, also known as deterministic letter-to-letter transducers. More formally we have the following definition.

**Definition 3.5.** A (deterministic) Mealy machine is a 6-tuple \( (Q, q_0, \Delta, \Gamma, \delta, \theta) \) where \( Q \) is the finite set of states, \( q_0 \in Q \) is the initial state, \( \Delta \) is the input alphabet, \( \Gamma \) is the output alphabet, \( \delta : Q \times \Delta \to Q \) is the (partial) transition function, and \( \theta : Q \times \Delta \to \Gamma \) is the (partial) output function.

The concatenation of the output letters on a run of the underlying automaton provides a unique output word. The definition of the (partial) function \( \Delta^\omega \to \Gamma^\omega \) realized by the machine follows directly; we do not care about acceptance.

**Proposition 3.6.** Let \( \alpha \in \Gamma^\omega \) and \( \beta \in \Delta^\omega \). Then \( \text{Reg}[\alpha] \subseteq \text{Reg}[\beta] \) holds if and only if \( \alpha \) is the image of \( \beta \) under some function realized by a Mealy machine.

**Proof.** A Mealy machine answering \( \alpha \) on \( \beta \) clearly provides a way to transform any language of \( \text{Reg}[\alpha] \) into a language of \( \text{Reg}[\beta] \).

Conversely, if \( \text{Reg}[\alpha] \subseteq \text{Reg}[\beta] \) then \( \text{Pref}(\alpha) = \{ w \mid w \otimes \beta : |w| \in \mathcal{L}(A) \} \) for some deterministic automaton \( A \) on \( \Gamma \times \Delta \). Since the (unique) run on a finite word \( \alpha : n \otimes \beta : n \) only uses accepting states (due to determinism and prefix-closure), non-accepting states can wlog be removed. If the resulting automaton has transitions of the form \( q \to^{(a,b)} q' \) and \( q \to^{(a',b)} q'' \) with \( a \neq a' \), we can remove them. Indeed, it is impossible for a run on some \( \alpha : n \otimes \beta : n \) to use one of them, since all states are now accepting. This last automaton can easily be seen as a Mealy machine.

Comparison via \( \preceq_{\text{Reg}} \) thus corresponds to computability via Mealy machines. The properties of this preorder were studied under this form in [Bel08]. However, tiny changes in the words completely modify their \( \preceq_{\text{Reg}} \)-degree (see Fact 3.7): those classes are far from being robust.

**Fact 3.7 ([Bel08]).** Let \( \alpha \) be any non-ultimately periodic word.

(i) \( \alpha \prec_{\text{Reg}} \alpha : 1 : \prec_{\text{Reg}} \cdots \prec_{\text{Reg}} \alpha : n : \prec_{\text{Reg}} \cdots \) is a strictly increasing chain;

(ii) \( \alpha \succ_{\text{Reg}} \Box \alpha \succ_{\text{Reg}} \cdots \succ_{\text{Reg}} \Box^n \alpha \succ_{\text{Reg}} \cdots \) is a strictly decreasing chain.

Another interesting point is the closure properties of these classes. If regular languages are preserved under almost every reasonable operations, it can easily be shown that \( \text{Reg}[\alpha] \) keeps the most basic of them.

**Proposition 3.8 ([BS69]).** \( \text{Reg}[\alpha] \) is closed under boolean operations.

However, when \( \alpha \) is not ultimately periodic, \( \text{Reg}[\alpha] \) is not closed under projection (with respect to \( \otimes \)) [Rei13]. This is a serious issue if one intends to encode logical theories, what may explain why automata with advice have remained unused for many years. A possible solution, detailed in the next paragraph, is to use \( \omega \)-regularity instead of finite regularity.
3.2 Non-terminating languages and $\omega$-regularity

Once more, we shall provide a definition in terms of languages, but it could equivalently be defined with $\omega$-automata that read an advice string.

**Definition 3.9** ([KRSZ12]). $L \subseteq \Sigma^\omega$ is $\omega$-regular with advice $\alpha \in \Gamma^\omega$ if there is an $\omega$-regular language $L' \subseteq (\Sigma \times \Gamma)^\omega$ such that $L = \{ w \mid w \otimes \alpha \in L' \}$.

**Example 3.10.**

(i) Every $\omega$-regular language is also $\omega$-regular with any advice;

(ii) $\{ \alpha \}$ is $\omega$-regular with advice $\alpha$.

We denote by $\omega{\sf Reg}[\alpha]$ the class of $\omega$-regular languages with advice $\alpha$. The next definition generalizes $\omega$-regularity with advice to finite word languages.

**Definition 3.11** ([KRSZ12]). A language $L \subseteq \Sigma^*$ is non-terminating regular with advice $\alpha \in \Gamma^\omega$ if there is an $\omega$-regular language $L' \subseteq ((\Sigma \cup \Box) \times \Gamma)^\omega$ such that $L = \{ w \mid w \otimes \alpha \in L' \}$.

**Example 3.12.** $\forall n \geq 0$, $\text{Pref}(\alpha[n:])$ is non-terminating regular with advice $\alpha$.

Let $\text{Reg}^\infty[\alpha]$ the class of non-terminating regular languages with advice $\alpha$. The reader may notice that $L \in \text{Reg}^\infty[\alpha] \iff \{ w\Box^\omega \mid w \in L \} \in \omega{\sf Reg}[\alpha]$. In other words, a non-terminating automaton is allowed to check $\omega$-regular properties on suffixes of the advice.

**Fact 3.13** ([KRSZ12]). $\forall \alpha \in \Gamma^\omega$, $\text{Reg}^\infty[\alpha] \subseteq \text{Reg}^\infty [\alpha]$ and the inclusion is strict if (and only if) $\alpha$ is not ultimately periodic.

These new definitions thus increase the expressiveness of advice languages. Furthermore, they solves the issue concerning closure properties raised in the end of Section 3.1. This is shown in Proposition 3.14.

**Proposition 3.14** ([KRSZ12]). $\text{Reg}^\infty[\alpha]$ and $\omega{\sf Reg}[\alpha]$ are closed under boolean operations, cylindrification, and projection (with respect to $\otimes$).

Let us compare infinite words with respect to this $\omega$-regular use of advice. We define the preorders $\preceq_{\text{Reg}^\infty}$ (resp. $\preceq_{\omega{\sf Reg}}$) based on the inclusion of the $\text{Reg}^\infty$ (resp. $\omega{\sf Reg}$) classes, and the corresponding notions of degree similarly to what was done in Subsection 3.1 for $\preceq_{\text{Reg}}$.

**Fact 3.15.** Ultimately periodic words are the least $\preceq_{\text{Reg}^\infty}$- and $\preceq_{\omega\text{Reg}}$-degree.

We make another step towards a generic correspondence between advices, machine transductions, and logic.

**Definition 3.16.** An $\omega$-regular function $f$ is a (partial) mapping $\Gamma^\omega \to \Delta^\omega$ whose graph $\{ w \otimes f(w) \mid w \in \text{dom}(f) \}$ is an $\omega$-regular language.

**Definition 3.17.** We say that $\alpha \in \Gamma^\omega$ is the image of $\beta \in \Delta^\omega$ under an $\text{MSO}$-relabelling if there is a tuple $\text{MSO}^\omega[\prec, \Delta]$-formulas $(\phi_a(x))_{a \in \Gamma}$ such that: $\forall n \geq 0$, $\alpha[n] = a$ if and only if $\beta \models \phi_a(n)$.

**Proposition 3.18.** For $\alpha \in \Gamma^\omega$ and $\beta \in \Delta^\omega$, the following are equivalent:

(i) $\text{Reg}^\infty[\alpha] \subseteq \text{Reg}^\infty[\beta]$;
(ii) \( \omega \text{Reg}[\alpha] \subseteq \omega \text{Reg}[\beta] \);

(iii) \( \alpha \) is the image of \( \beta \) under some \( \omega \)-regular function;

(iv) \( \alpha \) is the image of \( \beta \) under some MSO-relabelling.

Proof. See Appendix A.

Remark 3.19. Mealy machines describe particular \( \omega \)-regular functions. In a logical point of view, those machines describe particular MSO-relabelings that cannot read the future, what will be formalized in Fact 4.20.

We thus obtain \( \approx_{\omega \text{Reg}} = \approx_{\text{Reg}^\infty} \) and \( \approx_{\text{Reg}} \subseteq \approx_{\text{Reg}^\infty} \), but this last inclusion is strict: an explicit difference is pointed out between Fact 3.7 and Example 3.12.

In order to understand the limits of the preorder \( \approx_{\text{Reg}^\infty} \), we briefly give a simple necessary condition for \( \alpha \approx_{\text{Reg}^\infty} \beta \).

Definition 3.20. Let \( \alpha \in \Gamma^\omega \). Its subword complexity is the function \( p_\alpha : \mathbb{N} \to \mathbb{N} \) defined by
\[
p_\alpha(k) = \#\{w \in \Gamma^k \mid w \text{ subword of } \alpha\}.
\]

This function counts for each \( k \geq 0 \) the number of factors of size \( k \) appearing in \( w \). We now show that this measure can only decrease when applying an \( \omega \)-regular function, or equivalently using the preorder \( \approx_{\text{Reg}^\infty} \).

Proposition 3.21. If \( \alpha \in \Gamma^\omega \) is the image of \( \beta \in \Delta^\omega \) under some \( \omega \)-regular function, then
\[
p_\alpha(k) \leq K(p_\beta(k))
\]
for some constant \( K \geq 0 \).

Proof. Let \( \mathcal{A} \) be an \( \omega \)-automaton recognizing the graph in \( (\Gamma \times \Delta)^\omega \) of the \( \omega \)-regular function, then \( \{\alpha\} = \{w \mid w \odot \beta \in \mathcal{L}(\mathcal{A})\} \). Let \( \rho \) be an accepting run of \( \mathcal{A} \) on \( \alpha \odot \beta \), \( \rho(m) \) being state after reading letter \( m \). If \( k, m, m' \geq 0 \) are such that
\[
\beta[m : m + k] = \beta[m' : m' + k], \ \rho(m - 1) = \rho(m' - 1), \ \text{and} \ \rho(m + k) = \rho(m' + k),
\]
then both \( \alpha \odot \beta \) and \( (\alpha[\cdot : m][\alpha[m' : m' + k][\alpha[m + k + 1 : ]] \odot \beta \) are accepted. Thus \( \alpha[m : m + k] \) and \( \alpha[m' : m' + k'] \) must be equal. This shows that \( \forall k \geq 0, p_\alpha(k) \leq |Q|^2 \times p_\beta(k) \) where \( Q \) is the set of states of \( \mathcal{A} \).

Ultimately periodic words can be characterized as strings with bounded subword complexity (see e.g. [AS03] for more details). This result is consistent with the fact that they form the least \( \approx_{\text{Reg}^\infty} \)-degree.

Remark 3.22. For all \( n \geq 1 \), concatenating all the strings of \( \{1, \ldots, n\}^* \) produces an infinite word \( \alpha_n \) such that \( p_\alpha(k) = n^k \). Then \( \text{Reg}^\infty[\alpha_n] \) is not contained in any \( \text{Reg}^\infty[\beta] \) for \( \beta \in \{1, \ldots, n - 1\}^\omega \) because \( p_\beta(k) \leq (n - 1)^k \).

This last remark remains somehow frustrating. Indeed, it shows that the size of the alphabet is an unavoidable parameter for \( \approx_{\text{Reg}^\infty} \). This is not good news when looking for a robust notion of complexity. The rest of this paper is goes towards the study of a more relevant preorder, based on presentable structures and not directly on languages.
4 Advice automatic structures

The interesting point of Section 3 is the philosophy raised by Propositions 3.6 and 3.18: they relate the power of advices to logical or computational comparisons. However, the resulting preorders were somehow disappointing.

We now turn to classes of presentable structures with advice in order to derive similar results. Following the definitions of [AGR17], we denote by $\text{AutStr}[\alpha]$ the class of $\text{Reg}[\alpha]$-presentable structures, $\text{AutStr}^\infty[\alpha]$ for $\text{Reg}^\infty[\alpha]$-presentable, and $\omega\text{AutStr}[\alpha]$ for $\omega\text{Reg}[\alpha]$-presentable (see Definition 2.2). Such structures are said (\(\omega\)-)automatic with advice \(\alpha\). Their study is located a level of abstraction higher than what was done in Section 3, since the languages have no longer importance in theirselves, but are only used to encode other objects.

We establish basic properties of presentations with advice and recall the main arguments that motivated their study. A more involved result is then given, showing that $\text{AutStr}[\alpha] = \text{AutStr}^\infty[\alpha]$ for every advice (Corollary 4.17). Section 5 will be devoted to the study of the preorders of infinite words.

4.1 Tools and basic properties of advice presentations

Fact 4.1. Inclusion of language classes give $\text{AutStr} \subseteq \text{AutStr}[\alpha] \subseteq \text{AutStr}^\infty[\alpha]$ and $\omega\text{AutStr} \subseteq \omega\text{Reg}[\alpha]$. Inclusions are equalities if $\alpha$ is ultimately periodic.

Remark 4.2. There is no immediate argument to deduce $\text{AutStr} \subsetneq \text{AutStr}[\alpha]$ from $\text{Reg} \subsetneq \text{Reg}[\alpha]$. We shall see in Section 7 that this statement is true.

As an immediate consequence of the definitions, $\text{AutStr}^\infty[\alpha] \subseteq \omega\text{AutStr}[\alpha]$ and $\omega\text{AutStr}[\alpha]$ contains uncountable structures (like uncountable sets with empty signature), whereas $\text{AutStr}^\infty[\alpha]$ does not. The relationship between these two classes is made explicit when refining this idea.

Theorem 4.3 ([AGR17]). $\text{AutStr}^\infty[\alpha]$ is exactly the subclass of countable structures of $\omega\text{AutStr}[\alpha]$.

Using logical descriptions of advice automatic structures will be helpful in the sequel. The next result generalizes the case of automatic structures and shows how the advice contains the seeds of every automatic presentation. Its proof makes use of the closure properties detailed in the previous section.

Proposition 4.4 ([Abu16]).
(i) $\mathfrak{A} \in \omega\text{AutStr}[\alpha]$ if and only if $\mathfrak{A}$ is $S$-interpretable in $\mathcal{M}^\alpha$;
(ii) $\mathfrak{A} \in \text{AutStr}^\infty[\alpha]$ if and only if $\mathfrak{A}$ is $FS$-interpretable in $\mathcal{M}^\alpha$.

Remark 4.5. If the presentation is injective, so is the resulting interpretation.

Remark 4.6. If $\mathfrak{A}$ is presented over a binary encoding alphabet $\{0, 1\}$, the resulting interpretation can be done 1-dimensional.

We note that an advice automatic structure can be “effectively” described via a tuple of automata (as for standard automatic structures), plus a certain advice $\alpha$. In fact, the decidability feature of automatic structures is preserved as soon as $\alpha$ is decidable enough (and the proof is very similar).
Theorem 4.7 ([AGR17]).

(i) AutStr^∞[α] and ωAutStr[α] are closed under FO-interpretations;

(ii) If $\mathcal{W}^\alpha$ has a decidable MSO-theory, every structure in ωAutStr[α] (thus in AutStr[α] as well) has a decidable FO-theory.

Large classes of infinite words with decidable MSO-theory have been described, see e.g. [Bar08], [Sem84] or [RT06]. We briefly show why the generalization from automatic structures to advice automatic structures can be fruitful (compare the next result to Theorem 2.5).

Fact 4.8 ([KRZ12]). $⟨\mathbb{Q},+⟩ \in$ AutStr[α] for an α with decidable MSO-theory.

We now discuss a few structural properties of advice automatic structures. The statements are not deeply technical nor enlightening, but they are essential tools in the discussions of Section 5. A first question is to know whether each presentation can be made injective.

Proposition 4.9 ([Rei13]). If A has a $\text{Reg}^\infty[α]$-presentation, it has an injective $\text{Reg}^\infty[μ_n(α)]$-presentation.

A second point is to understand how the encoding alphabet can be restricted. Binary presentations are enough to describe all automatic structures [BG00]. We show that it is still possible here, up to a small modification of the advice.

Definition 4.10. For $n \geq 1$ let $μ_n : \Gamma \to \Gamma^*$ mapping each letter $a$ to $a^n$. We extend this function to infinite words in a morphic way.

Example 4.11. $μ_3((01)\omega) = (000111)\omega$.

Proposition 4.12. If A has a $\text{Reg}^\infty[α]$-presentation, there is $n \geq 1$ such that A has a $\text{Reg}^\infty[μ_n(α)]$-presentation over a binary encoding alphabet. If the first presentation was injective, so in the second.

Proof sketch. Let $A = (A,R_1,\ldots,R_n) \in \text{AutStr}^\infty[β]$ and $(L, L_0, L_1, \ldots, L_n)$ the corresponding presentation over an alphabet $Σ = \{a_1 \ldots a_n\}$. The idea is to replace $a_i$ by a binary string of length $n$. Formally let $w_i = 0^k1^{n-k}$ and let $f : Σ \to \{0,1\}$ mapping $a_i$ to $w_i$. $f$ is extended morphically to (convolutions of) words of $Σ^*$. Note $|f(w)| = n|w|$. We check that $(f(L), f(L_0), f(L_1), \ldots, f(L_n))$ is a tuple of languages of $\text{Reg}^\infty[μ_n(β)]$ which is still a presentation of $A$. If $L_w = \{w \otimes w \mid w \in L\}$ then $f(L_w) = \{w \otimes w \mid w \in f(L)\}$ thus injectivity is preserved by this construction.

Remark 4.13. This proof also works for $\text{Reg}[α]$- and $ω\text{Reg}[α]$-presentations.

4.2 Terminating and non-terminating encodings

We did not deal much with $\text{Reg}[α]$-presentations. Indeed, since useful closure properties lack to this class of languages, the situation seems more difficult. A natural problem is however to find out whether every structure in AutStr^∞[α] has such a terminating presentation. Another formulation of this question is: does AutStr[α] = AutStr^∞[α] hold? We give a positive answer.

To understand how the proof works, we first need to explicit the main difference between $\text{Reg}[α]$ and $\text{Reg}^∞[α]$. An $ω$-automaton performs an infinite run
on $w \otimes \alpha$ (for $w$ finite) in two steps: first, it follows a finite run on $w \otimes \alpha[\cdot \mid w]$, then it checks an $\omega$-regular property on $\exists^{\omega} \otimes \alpha[\cdot \mid w] \simeq \alpha[\cdot \mid w]$. Basically, it only uses the $\omega$-regularity feature on suffixes of the advice. On the other hand, a terminating automaton is blind to the $\omega$-future. We show that it can nevertheless look at some “finite amount of future” and deduce corresponding $\omega$-regularity on the suffixes. A key idea is that since the advice is fixed, so are several properties of its suffixes. This is detailed in Theorem 4.14.

**Theorem 4.14.** Let $L$ be an $\omega$-regular language and $\alpha \in \Gamma^\omega$ a fixed word. There is a (finite words) regular language $L'$ and $N \geq 0$ such that for all $n \geq N$ we have $\alpha[n \cdot] \in L$ if and only $\alpha[n \cdot]$ has a finite prefix in $L'$. Furthermore, if $L$ can be described by a $\text{FO}[<, \Gamma]$-sentence, $L'$ can be described by a $\text{FO}[<, \Gamma]$-sentence.

**Proof idea.** Both proofs are detailed in Appendix B. The case of $\text{FO}$ is achieved via expressive equivalence with LTL (known as Kamp’s Theorem, see [Rab14]). For $\text{MSO}$ in general, we make use of results of A.L. Semenov [Sem84].

We shall not make use of $\text{FO}[<, \Gamma]$ alone in the sequel. In order to give a more readable statement in our context, let $\otimes$ be a new padding symbol.

**Corollary 4.15.** Let $L \subseteq \Gamma^\omega$ be an $\omega$-regular language and $\alpha \in \Gamma^\omega$. There is a function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that $\{0^n \otimes f(n) \mid \alpha[n \cdot] \in \text{Reg}[\alpha]\}$.

**Proof.** By applying Theorem 4.14 we get a regular language $L'$ and $N \geq 0$ such that for all $n \geq N$, $\alpha[n \cdot] \in L$ if and only if $\alpha[n \cdot]$ has a finite prefix in $L'$. If $\alpha[n \cdot] \in L$, let $f(n)$ be the length of the smallest prefix of $\alpha[n \cdot]$ belonging to $L'$. We take $f(n)$ arbitrarily in the other cases to define a mapping $f : \mathbb{N} \rightarrow \mathbb{N}$. The set $\{0^n \otimes f(n) \mid n \geq N$ and $\alpha[n : n + f(n) + 1 \mid L'] = \{0^n \otimes f(n) \mid n \geq N$ and $\alpha[n \cdot] \in L\}$ is clearly terminating regular with advice $\alpha$. Thus $\{0^n \otimes f(n) \mid n \geq 0$ and $\alpha[n \cdot] \in L\} \in \text{Reg}[\alpha]$ as well (we hardcoded in the automaton what happens before $N$).

**Corollary 4.15** formalizes our intuition that terminating automata can check $\omega$-regular properties on suffixes. We can now detail the relationship between $\text{Reg}[\alpha]$ and $\text{Reg}^\omega[\alpha]$ as a transformation that adds padding symbols $\otimes$ to the words of the language.

**Corollary 4.16.** Let $\alpha \in \Gamma^\omega$. For every language $L \in \text{Reg}^\omega[\alpha]$, there is a function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that $\{w \otimes f(|w|) \mid w \in L\} \in \text{Reg}[\alpha]$.

**Proof sketch.** $L = \{w \in \Sigma^* \mid w \otimes \alpha \in \mathcal{L}(\mathcal{A})\}$ for some $\omega$-automaton $\mathcal{A}$. This automaton checks the belonging of suffixes of $\alpha$ to a finite number of $\omega$-regular languages $L_1 \ldots L_n$ (as evoked in the beginning of this subsection). We take $f := \max f_1 \ldots f_n$ where each $f_i$ is the function given by Corollary 4.15 for $L_i$. The reader will get convinced that an automaton $\mathcal{A}'$ (for finite words) can be built so that $\{w \otimes f(|w|) \mid w \in L\} = \{v \mid v \otimes \alpha[\cdot \mid v] \in \mathcal{L}(\mathcal{A}')\}$.

The ideas developed above can be applied to get the desired result.

**Corollary 4.17.** For every advice $\alpha$, $\text{AutStr}[\alpha] = \text{AutStr}^\omega[\alpha]$.
Proof sketch. We follow the same sketch as for Corollary 4.15 and extend the \( \text{Reg}_\infty[\alpha]\)-presentation by adding a well-chosen finite number of padding symbols \( \perp \) behind each word. The function \( f \) is now a maximum over the properties of suffixes checked by all the \( \omega \)-automata for the languages of the presentation and the automata for their complements. Considering the complements is necessary since we also need to know when a property does not hold.

Remark 4.18. The term automatic structure with advice \( \alpha \) can thus be applied without ambiguity to a structure in \( \text{AutStr}[\alpha] \) or in \( \text{AutStr}_\infty[\alpha] \).

4.3 Digression: relativization of MSO-formulas

The results of this subsection can be considered as a digression since they will not be helpful for the rest of our study. As an application of Theorem 4.14, we provide an original normal form for MSO-formulas with free variables when interpreted in a fixed word model. We first take some abbreviations for MSO\(<,\Gamma\)-formulas:

\[
\forall x \leq y \phi \quad \text{stands for} \quad \forall x (x \leq y \rightarrow \phi)
\]

and

\[
\exists x \leq y \phi \quad \text{for} \quad \exists x (x \leq y \land \phi).
\]

Similarly with set quantifications:

\[
\forall X \leq y \phi \quad \text{for} \quad \forall X ((\forall x (x \in X \rightarrow x \leq y)) \rightarrow \phi)
\]

and

\[
\exists X \leq y \phi \quad \text{for} \quad \exists X ((\forall x (x \in X \rightarrow x \leq y)) \land \phi).
\]

Definition 4.19 (relativized formulas). A MSO\(<,\Gamma\)-formula \( \phi \) with free variables \( X, x, y \) is said to be relativized under \( y \) if

\[
\phi(X, x, y) = \bigwedge_{x \in \mathcal{P}} x \leq y \land \bigwedge_{X \in \mathcal{X}} X \leq y \land \psi(X, x)
\]

and every quantification in \( \psi \) is of the form \( Qz \leq y \) or \( QZ \leq y \).

We note that relativized sentences provide a suitable logical formalism to describe the transformations performed by Mealy machines. The proof of the next fact follows from standard logic-automata transformations.

Fact 4.20. \( \alpha \in \Gamma^\omega \) is the image of \( \beta \in \Delta^\omega \) under some Mealy machine if and only there exists a tuple of MSO\(<,\Delta\)-formulas \( (\phi_a(x))_{a \in \Gamma} \) relativized under \( x \), such that for all \( n \geq 0 \), \( \alpha[n] = a \) if and only if \( \beta \models \phi_a(n) \). We call such a tuple a relativized MSO-relabeling (compare with Definition 3.17).

We now consider formulas of the form \( \exists y \phi \) where \( \phi \) is relativized under \( y \). These formulas are far less expressive than full MSO. Indeed, when such a sentence holds in a word model, there is a finite proof of its validity.

Example 4.21. Let \( \phi := \forall x \exists y x > y \land P_a(y) \) meaning “there are infinitely many letters \( a \)”. There is no relativized sentence equivalent to \( \phi \), but among the suffixes \( \alpha[:n] \) of a fixed word \( \alpha \), this property either always or never holds.

We now show that such formulas (with free variables) are enough to describe the full power of MSO in a fixed infinite word model.

Corollary 4.22. Let \( \phi(X, x) \) be a MSO\(<,\Gamma\)-formula and \( \alpha \in \Gamma^\omega \) fixed. There is a formula \( \psi(X, x, y) \) relativized under \( y \) such that for every tuple \( \mathcal{A} \) of finite sets, and tuple \( \mathcal{P} \) of positions: \( \alpha \models \phi(X, x) \) if and only if \( \alpha \models \exists y \psi(X, x, y) \).
Proof sketch. We treat the case of formulas \( \phi(X) \) with one free set variable. If \( A \subseteq \mathbb{N} \) is a finite set, denote by \( \chi_A \in \{0,1\}^* \) the finite word of length \( \max A + 1 \) with \( \chi[n] = 1 \) iff \( n \in A \). It follows from standard logic-automata translations that \( \{ \chi_A \mid A \text{ finite and } \alpha \vdash \phi(A) \} \) is an \( \omega \)-regular language, thus \( \{ \chi_A \mid A \text{ finite and } \alpha \vdash \phi(A) \} \in \text{Reg}_\omega[\alpha] \). From Theorem 4.11 we get that \( L := \{ \chi_A \mid A \text{ finite and } \alpha \vdash \phi(A) \} \in \text{Reg}[\alpha] \) for some \( f : \mathbb{N} \rightarrow \mathbb{N} \). Hence there is a finite words automaton \( A \) such that \( L = \{ w \mid w \in \text{dom}(\alpha) \} \). It can be translated back into a formula \( \exists y X \leq y \land \psi(X,y) \) with restricted quantifications, where \( X \) describes the possible set of positions labelled by 1.

Remark 4.23. Similar statements may be described for \( \text{FO} \) alone.

5 Transductions and advice presentations

After the fundamental results of Section 4 on advice automatic structures, we are now able to understand which preorder they describe over infinite words. Corollary 4.14 implies in particular that \( \text{AutStr}[\alpha] \subseteq \text{AutStr}[\beta] \) if and only if \( \text{AutStr}_\omega[\alpha] \subseteq \text{AutStr}_\omega[\beta] \). The objective of this section is to show equivalence with \( \omega \text{AutStr}[\alpha] \subseteq \omega \text{AutStr}[\beta] \) and give several other characterizations. The climaxes lie in Theorem 5.12 and Theorem 5.21, where we relate our notions to well-known logical transformations and finite transducers.

5.1 MSO-transductions

First, we recall in this subsection the definition of a logical tool that can be seen as a particular form of MSO-interpretation.

Definition 5.1. A \((k\text{-copying})\) MSO-transduction (MSOT) from \( \Delta^\omega \) to \( \Gamma^\omega \) is a tuple of \( \text{MSO}^\omega \)-formulas with free first-order variables.

\[
(\phi_1^\alpha(x))_{a \in \Gamma} \ldots (\phi_k^\alpha(x))_{a \in \Gamma}, (\phi_{i,j}^\alpha(x,y))_{1 \leq i,j \leq k}
\]

The semantic of an MSOT \( \tau \) is defined as that of an MSO-interpretation in \( k \) disjoint copies of a host word structure. More precisely, the structure \( I_\tau(\mathcal{M}^\beta) \) (not necessarily a word) has signature \( \{<,(P_a)_{a \in \Gamma}\} \) and is defined as follows:

- \( \text{dom}(I_\tau(\mathcal{M}^\beta)) = \bigcup_{1 \leq i \leq k} \{ (n,i) \mid \text{there is a } \alpha \text{ such that } \beta \models \phi_\alpha(n) \} \);
- if \( (n,i) \in \text{dom}(I_\tau(\mathcal{M}^\beta)) \), then \( (n,i) \in P_a \) if and only if \( \beta \models \phi_\alpha(n) \);
- if \( (m,j) \in \text{dom}(I_\tau(\mathcal{M}^\beta)) \), then \( (n,i) < (m,j) \) if and only if \( U \models \phi_{i,j}^\alpha(n,m) \).

Since we are interested in transformations between words, we only consider the case when \( I_\tau(\mathcal{M}^\beta) \) is a word structure (what is syntactically definable by adding a domain \( \text{MSO}^\omega \)-sentence). Each MSO-transduction \( \tau \) realizes now a \((k\text{-partial})\) function \( \tau : \Delta^\omega \rightarrow \Gamma^\omega \) whose domain is \( \{ \beta \in \Delta^\omega \mid I_\tau(\mathcal{M}^\beta) \text{ is (isomorphic to) a word structure} \} \), the image \( \tau(\beta) \) of \( \beta \) being the unique \( \alpha \) such that \( I_\tau(\mathcal{M}^\beta) = \mathcal{M}^\alpha \).

The reader is asked to keep in mind that MSOT define a certain class of functions on infinite strings, even if our main concern is only the existence of a transduction between two fixed words. We write \( \alpha \preceq_{\text{MSOT}} \beta \) if there is a MSO-transduction \( \tau \) such that \( \tau(\beta) = \alpha \).
Remark 5.2. Even if MSO-interpretations in general are not closed under composition (Remark 2.9), it is the case of MSOT (see e.g. [AFT12], the problem of tuples of sets disappears). Thus $\leq_{\text{MSOT}}$ is transitive, and is even a preorder over infinite words. Furthermore, the composition of an MSOT and a $S$-interpretation can be realized by an unique $S$-interpretation.

Remark 5.3. The (relativized) MSO-relabelings (Definition 3.17 and Fact 4.20) can be seen as syntactical fragments of 1-copying MSOT. Hence $\leq_{\text{MSOT}}$ is a more generic notion of comparison than the preorders of Section 3; it will be noticed in Remark 6.2 that the increase of power is strict.

Example 5.4. (i) If $\alpha \leq_{\text{Reg}} \beta$ then $\alpha \leq_{\text{MSOT}} \beta$ (Remark 5.3); (ii) modifying a finite part of $\alpha$ does not change its MSOT-degree; (iii) if the $\mu_n$ are the morphisms of Definition 4.10 then $\mu_n(\alpha) \leq_{\text{MSOT}} \alpha$ and $\alpha \leq_{\text{MSOT}} \mu_n(\alpha)$ for all $n \geq 1$; (iv) if $w$ is a finite word, denote by $\tilde{w}$ its mirror image. If $\alpha := w_1\#w_2\#\cdots \in (\Gamma^*\#)^\omega$, we define $\tilde{\alpha} := \tilde{w_1}\#\tilde{w_2}\#\cdots$. Then $\tilde{\alpha} \leq_{\text{MSOT}} \alpha$ since the new ordering of the letters is MSO-definable; the behavior of the MSOT is described graphically in Figure 1.

\[\alpha := a \rightarrow b \rightarrow \# \rightarrow b \rightarrow a \rightarrow a \rightarrow \# \rightarrow \cdots\]

\[\tilde{\alpha} = a \rightarrow b \rightarrow \# \rightarrow b \rightarrow a \rightarrow a \rightarrow \# \rightarrow \cdots\]

Figure 1: Reversing the factors with an MSOT

5.2 From automatic structures to logical transductions

We forget one minute the definition of MSOT and come back to advice automatic structures. When looking for some complete structure of an advice, a simple idea would be that $\mathfrak{A} \in \text{AutStr}^\infty[\alpha]$ if and only if $\text{AutStr}^\infty[\alpha] \subseteq \text{AutStr}^\infty[\beta]$. However, this statement will turn out to be wrong (see Remark 6.6). We need a stronger object that is presented in Definition 5.5.

Definition 5.5 ([LC07]). Let $\mathfrak{A} = \langle A, R_1, \ldots, R_n \rangle$ be a structure, we define its weak powerset structure $\mathcal{P}^f(\mathfrak{A})$ as the structure $(\mathcal{P}^f(A), R'_1, \ldots, R'_n, \subseteq)$ where:

- $\mathcal{P}^f(A)$ is the weak powerset (set of finite subsets) of $A$;
- $\subseteq$ is the inclusion relation on $\mathcal{P}^f(A)$;
- $R'_i(A_1, \ldots, A_r)$ holds in $\mathcal{P}^f(\mathfrak{A})$ if and only if $A_1, \ldots, A_r$ are singletons $\{a_1\}, \ldots, \{a_r\}$ and $R_i(a_1, \ldots, a_r)$ holds in $\mathfrak{A}$.

Fact 5.6. $\mathfrak{A}$ is $FS$-interpretable in $\mathfrak{B}$ if and only if $\mathfrak{A}$ is $\text{FO}$-interpretable in $\mathcal{P}^f(\mathfrak{B})$. Thus by Proposition 4.4, $\text{AutStr}^\infty[\alpha]$ is the class of structures $\text{FO}$-interpretable in $\mathcal{P}^f(\mathfrak{B})$. 

15
Proposition 5.7. For all infinite words $\alpha$, $\beta$, $\text{AutStr}^\infty[\alpha] \subseteq \text{AutStr}^\infty[\beta]$ if and only if $\mathcal{P}^f(\mathcal{W}^\alpha) \in \text{AutStr}^\infty[\beta]$.

Proof. If $\text{AutStr}^\infty[\alpha] \subseteq \text{AutStr}^\infty[\beta]$ then $\mathcal{P}^f(\mathcal{W}^\alpha) \in \text{AutStr}^\infty[\beta]$, since clearly $\mathcal{P}^f(\mathcal{W}^\alpha) \in \text{AutStr}^\infty[\alpha]$. Conversely, if $\mathcal{P}^f(\mathcal{W}^\alpha) \in \text{AutStr}^\infty[\beta]$, then $\mathcal{P}^f(\mathcal{W}^\alpha)$ is FO-interpretable in $\mathcal{P}^f(\mathcal{W}^\beta)$. If $\mathcal{A} \in \text{AutStr}^\infty[\alpha]$, $\mathcal{A}$ is FO-interpretable in $\mathcal{P}^f(\mathcal{W}^\beta)$ and we get that $\mathcal{A}$ is FO-interpretable in $\mathcal{P}^f(\mathcal{W}^\beta)$ because FO-interpretations are closed under composition. Thus $\mathcal{A} \in \text{AutStr}^\infty[\beta]$.

This result a provides characterization which is quite abstract and, in some respects, trivial. Nevertheless, we get the intuition that powerset structures are a key notion to understand advice automaticity. In the next theorem, a ($\Delta$-labelled) tree structure has the form $(A, <, (P_a)_{a \in \Delta})$ where the domain $A$ is a prefix-closed subset of $\{0, 1\}^*$, $w < w'$ holds whenever $w$ is a prefix of $w'$ and the $P_a$ label the nodes of $A$ with $\Delta$.

Theorem 5.8 ([LC07], Corollary 4.4). Let $\mathcal{A}$ a structure and $\mathcal{T}$ a tree structure. If $\mathcal{P}^f(\mathcal{A})$ is 1-dimensionally injectively FS-interpretable in $\mathcal{T}$, then $\mathcal{A}$ is 1-dimensionally injectively WMSO-interpretable in $\mathcal{T}$.

In the case of advice automatic structures, Theorem 5.8 is at the same time too generic and too restrictive. On the one hand, we only consider interpretations in word structures $\mathcal{W}^n$, which are very specific trees. On the other hand, we need arbitrarily dimensional FS-interpretations, and they are not supposed to be injective. Hopefully propositions 4.9 and 4.12 given in Section 4 drive us towards these required conditions, with a small change.

Corollary 5.9. If $\mathcal{P}^f(\mathcal{W}^\alpha) \in \text{AutStr}^\infty[\beta]$, then $\alpha \preceq_{\text{MSOT}} \beta$.

Proof. Assume the hypothesis holds. Then by propositions 4.9 and 4.12 $\mathcal{P}^f(\mathcal{W}^\alpha)$ has an injective $\text{Reg}^\infty[\mu_n(\beta)]$-presentation over a binary alphabet, for some $n \geq 1$. Thus $\mathcal{P}^f(\mathcal{W}^\alpha)$ is 1-dimensionally injectively FS-interpretable in $\mathcal{W}^{\mu_n(\beta)}$ (remarks 1.5 and 1.6). By applying Theorem 5.8 $\mathcal{W}^\alpha$ is 1-dimensionally injectively WMSO-interpretable in $\mathcal{W}^{\mu_n(\beta)}$. Such interpretations are a particular case of MSOT, so $\alpha \preceq_{\text{MSOT}} \mu_n(\beta)$. Since $\mu_n(\beta) \preceq_{\text{MSOT}} \beta$ (Example 5.3), composing both transductions provides $\alpha \preceq_{\text{MSOT}} \beta$.

Since the proof Theorem 5.8 in [LC07] is rather long and involved, we provide in Appendix C a direct and self-contained proof of Corollary 5.9. It is largely based on their arguments, but avoids useless work in the specific case of infinite words and arbitrarily dimensional interpretations.

Remark 5.10. Corollary 5.9 can be extended to presentations using tree languages with (infinite) tree advice (see e.g. [AGRT17] for a definition), but this direction will not be followed in our study of infinite words.

Remark 5.11 (uniformly automatic classes). Let $P$ a set of infinite words. A class of structures $\mathcal{C}$ (over a given signature) is said uniformly automatic with advice set $P$ if there are fixed automata whose languages with advice $\alpha$ describe

\[\text{nothing was done but standard logic-automata transformations}\]
presentations of each structure in \(C\) when \(\alpha\) ranges in \(P\) \cite{AGRI17}. In particular, if \(P\) is \(\omega\)-regular, the \(\mathsf{FO}\)-theory of the class \(C\) is decidable. Since the proof of Theorem 5.8 only depend of the automata for the presentation of \(P^f(A)\), it can be generalized to show that if the uniform classes with \(P \subseteq \Gamma^\omega\) are also uniform with \(Q \subseteq \Delta^\omega\), then there is an \(\mathsf{MSO}\)-transduction \(\tau\) such that \(\tau(Q) = P\).

We now have all the ingredients to provide effortlessly a strong and elegant characterization for the inclusion of classes.

**Theorem 5.12.** The following conditions are equivalent:

(i) \(\omega\mathsf{AutStr}[\alpha] \subseteq \omega\mathsf{AutStr}[\beta]\);

(ii) \(\mathsf{AutStr}^\infty[\alpha] \subseteq \mathsf{AutStr}^\infty[\beta]\);

(iii) \(\alpha \subseteq_{\mathsf{MSOT}} \beta\).

**Proof.** Since being countable is an intrinsic property of structures, the way from (i) to (ii) is a consequence of Theorem 4.3. If (ii) holds, Proposition 5.7 and Corollary 5.9 give (iii). Finally if (iii) is true and \(A\) is \(S\)-interpretable in \(\mathcal{W}^\alpha\) (via Proposition 4.4), then \(A\) is \(S\)-interpretable in \(\mathcal{W}^\beta\), since the composition of an \(\mathsf{MSOT}\) and an \(S\)-interpretation can be rewritten as an \(S\)-interpretation.

Together with Corollary 4.17, this result described how all natural definitions for advice automatic structures converge towards the same comparison of infinite words, described by \(\mathsf{MSO}\)-transductions. This point gives a deep theoretical meaning to the resulting preorder. Another virtue of Theorem 5.12 is the ability to translate immediately the results of Example 5.4 in terms of advice automatic structures, as detailed in Example 5.13.

**Example 5.13.**

(i) If \(\alpha \subseteq_{\text{Reg}^\infty} \beta\) then \(\omega\mathsf{AutStr}[\alpha] \subseteq \omega\mathsf{AutStr}[\beta]\);

(ii) modifying a finite part of \(\alpha\) does not modify \(\omega\mathsf{AutStr}[\alpha]\);

(iii) \(\mathsf{AutStr}[\alpha] = \mathsf{AutStr}[\mu_n(\alpha)]\) for all \(n \geq 1\);

(iv) if \(\alpha \in (\Gamma^* \#)^\omega\), then \(\omega\mathsf{AutStr}[\alpha] = \omega\mathsf{AutStr}[\overline{\alpha}]\).

However, to sharpen our understanding of \(\subseteq_{\mathsf{MSOT}}\), we also need to understand when strict inclusions take place. Having an equivalent simple automata model to describe our transformations is very useful in that case.

### 5.3 An equivalent computational model

A two-way transducer is a finite deterministic two-way automaton with an additional one-way output tape. More formally we have the following definition.

**Definition 5.14.** A (deterministic) two-way finite transducer (2WFT) is a 6-tuple \((Q, q_0, \Delta \cup \{\#\}, \Gamma, \delta, \theta)\) where \(Q\) is the finite set of states, \(q_0 \in Q\) initial, \(\Delta\) input alphabet, \(\Gamma\) output alphabet, \(\delta : Q \times (\Delta \cup \{\#\}) \rightarrow Q \times \{\#, \}\) (partial) transition function, and \(\theta : Q \times (\Delta \cup \{\#\}) \rightarrow \Gamma^*\) (partial) output function.
The component $\{a,\triangleright\}$ determines the left or right move the reading head on a read-only input tape. When the 2WFT is given $\beta \in \Delta^\omega$ as an input word, this tape contains $\vdash \beta$ (adding a symbol $\vdash$ helps the transducer to notice the beginning of its input when going left). The definition of the (partial) function $\Delta^\omega \to \Gamma^\omega$ realized the 2WFT follows directly, as previously for Mealy machines.

**Remark 5.15.** The transducer is said to be one-way ($1WFT$, or just finite transducer) if all its transitions are of the form $(q,\triangleright)$. Mealy machines are a particular case of 1WFT whose input-output mechanism is one-to-one.

**Example 5.16.** There is a three-state 2WFT outputting $\tilde{\alpha}$ on every $\alpha \in (\Gamma\#)^\omega$. Its behavior is the following: scan a maximal $\#$-free block, read it in a reversed way while outputting, then output $\#$ and move to the next block.

Write $\alpha \lessdot_{2WFT} \beta$ if $\alpha$ is the image of $\beta$ under a function realized by some 2WFT. We now give some basic properties of these transductions.

**Fact 5.17.** If $\alpha$ is ultimately periodic, for every string $\beta$ we have $\alpha \lessdot_{2WFT} \beta$.

**Lemma 5.18.** Let $T$ be a 2WFT transforming $\beta$ into $\alpha$. If $\alpha$ is not ultimately periodic, there is an integer $N$ such that the run of $T$ does not visit more than $N$ times each position of the string $\vdash \beta$.

*Proof.* Let $N$ be the number of states of $T$. If $T$ visits more than $N$ times a position, it is caught in a loop and must output an ultimately periodic word.

When considering definable functions between finite strings, a well-known equivalence holds between MSOT and 2WFT (Theorem 5.19). The definitions of MSOT and 2WFT have to be slightly sharpened to get the exact correspondence, see details in [EH01].

**Theorem 5.19 ([EH01]).** (Partial) functions over finite words $\Delta^* \to \Gamma^*$ definable by MSOT are the (partial) functions realized by 2WFT.

Fairly recently, this result was extended to functions between infinite strings, but some complications quickly appear: deciding the validity of MSO-sentences is not always possible without reading the (variable) input entirely. Thus 2WFT alone are not powerful enough and they need extra features like $\omega$-regular lookahead, i.e. ability to check instantly $\omega$-regular properties of suffixes of the input. We shall not precise the definition, but Theorem 5.20 provides an overlook of the (semantical) conditions required.

**Theorem 5.20 ([AFT12]).** (Partial) functions over infinite words $\Delta^\omega \to \Gamma^\omega$ definable by MSOT are the (partial) functions realized by 2WFT with $\omega$-regular lookahead whose computations visit the whole input string.

When looking closely at Theorem 5.19 and Theorem 5.20 in the light of our previous results, a question arises naturally: it is possible to get rid of the lookaheads when fixing the input infinite word? Indeed, we have always considered transformation from a fixed word and we noticed in Section 4 that this restriction simplified certain notions. Theorem 5.21 gives a positive answer. This involved result is not a consequence of Theorem 5.20 since we are not aware of a manner to remove the lookaheads when fixing the input.
Theorem 5.21. \( \alpha \preceq_{\text{MSOT}} \beta \) if and only if \( \alpha \preceq_{\text{2WFT}} \beta \).

Proof idea. The way from 2WFT to MSOT is obvious if \( \alpha \) is ultimately periodic. If it is not the case, Lemma 5.18 shows that the run of the transducer must visit the whole input and we conclude by Theorem 5.20.

We show in Appendix D how to transform \( \alpha \preceq_{\text{MSOT}} \beta \) into \( \alpha \preceq_{\text{2WFT}} \beta \). The idea is to show how the equivalent model of streaming \( \omega \)-string transducer \[AFT12\] can be simulated by a 2WFT when fixing its input.

Remark 5.22. Without Theorem 5.21, it is not clear that \( \preceq_{\text{2WFT}} \) is transitive.

We finally remark that the definition of 2WFT can slightly simplified, what we will be very useful in the proofs of the next section.

Fact 5.23. We can consider w.l.o.g. in Theorem 5.21 that the transducer reads and moves on an input tape containing \( \beta \) instead of \( \top \beta \).

Proof. If \( \alpha \) is ultimately periodic, the result is obvious by Fact 5.17. Else, according to Lemma 5.18 there is \( N \geq 0 \) such that the transducer does not visit position 0 of \( \top \beta \) more than \( N \) times. Thus after a certain time \( N' \) the run never visits \( \top \). What it output before \( N' \) can be hardcoded, and the rest of the computation can be done on \( \beta \) directly.

6 The two-way transductions hierarchy

We initiate in this section a study of two-way transductions between infinite words. It can equivalently be seen as a study of the preorder defined by MSOT, or classes \( \text{AutStr}[\alpha] \), \( \text{AutStr}^\infty[\alpha] \) and \( \text{AutStr}[\alpha] \); but the 2WFT formulation is - as predicted above - the easiest way to deduce interesting statements. We shall use the term 2WFT hierarchy to describe the ordered set of 2WFT-degrees (i.e. equivalence classes of \( \preceq_{\text{2WFT}} \cap \succeq_{\text{2WFT}} \)).

A more or less similar work has been done in \[EKSW15\], with the relation \( \preceq_{1\text{WFT}} \) defined as follows: \( \alpha \preceq_{1\text{WFT}} \beta \) if and only if there is a one-way finite transducer writing \( \alpha \) on the input \( \beta \). This definition clearly describes a preorder. Even if no previous research has been carried on the 2WFT hierarchy, we shall see that several results on the 1WFT can be adapted in our context, after a variable amount of work. Note that \( \preceq_{1\text{WFT}} \subseteq \preceq_{\text{2WFT}} \).

Proposition 6.1.

(i) There are uncountably many distinct 2WFT-degrees;

(ii) a set of 2WFT-degree has an upper bound if and only if it is countable;

(iii) the 2WFT hierarchy has no greatest degree;

(iv) every 2WFT-degree contains a binary string.

Proof idea. The proofs of similar statements for 1WFT in \[EKSW15\] raise no specific issue and their adaptation is straightforward.
Remark 6.2. Considering binary strings is thus sufficient to describe all the degrees. Comparing this result with Proposition 3.21 shows that the preorder \( \preceq_{\text{Reg}} \) defined by MSO-relabelings is strictly weaker than \( \preceq_{\text{2WFT}} = \preceq_{\text{MSOT}} \).

As a consequence of Proposition 6.1, the 2WFT hierarchy is not trivial. We now show that it is fine-grained enough to distinguish ultimately periodic words.

Proposition 6.3. Ultimately periodic words are the least 2WFT-degree.

Proof. Assume that \( \alpha \) is ultimately periodic. Fact 5.17 concludes \( \alpha \preceq_{\text{MSOT}} \beta \) for all \( \beta \). Conversely, if \( \gamma \preceq_{\text{2WFT}} \alpha \), we show that \( \gamma \) is ultimately periodic. Since \( \alpha \preceq_{\text{2WFT}} \square \omega \) (take \( \beta = \square \omega \) in the previous argument), we get \( \gamma \preceq_{\text{2WFT}} \square \omega \).

Let \( T \) be the 2WFT computing this transformation and \((q_n, n_n) \geq 0 \) its run on \( \square \omega \) (sequence of tuples state/position). There exists \( j \geq 0 \) and \( k \geq 1 \) such that \( q_j = q_{j+k} \). Due to determinism and invariance of \( \square \omega \) by translation, this sequence of moves must to be repeated (note that necessarily \( n_{j+k} \geq n_j \)), and shows the ultimate periodicity of \( \gamma \).

Remark 6.4. Ultimately periodic words are also the least 1WFT degree.

This easy result shows, through the equivalences of Section 5, that non-trivial advices strictly increase the class of presentable structures (it was not obvious before). The next corollary concludes that we equivalently provided a characterization for automaticity of certain structures.

Corollary 6.5. \( P^f(2^{\mathbb{N}}) \) is automatic if and only if \( \alpha \) is ultimately periodic.

Proof. Relate Proposition 6.3 and Proposition 5.7 by the equivalences above.

Remark 6.6. There are structures \( 2^{\mathbb{N}} \) that are automatic when \( \alpha \) is not ultimately periodic \([\text{Bar08}]\), hence the automaticity of \( 2^{\mathbb{N}} \) is not equivalent to that of \( P^f(2^{\mathbb{N}}) \). One of the open questions in this field is to understand when \( 2^{\mathbb{N}} \) is automatic, and Corollary 6.5 may be considered as a small step in this direction.

We now turn to a more involved statement. A sequence \( \beta \) is said to be prime if it is a minimal but non-trivial word. Formally, \( \beta \) non ultimately periodic is prime in the 2WFT hierarchy if for all \( \alpha \preceq_{\text{2WFT}} \beta \), either \( \beta \preceq_{\text{2WFT}} \alpha \) or \( \alpha \) is ultimately periodic. The existence of prime sequences shows in particular that the 2WFT hierarchy is not dense.

Theorem 6.7. The sequence \( \pi := \prod_{n=0}^{\infty} 0^n1 \) is prime in the 2WFT-hierarchy.

Proof idea. We show in Appendix 3 that if \( \alpha \preceq_{\text{2WFT}} \pi \) then \( \alpha \preceq_{\text{1WFT}} \pi \), that is, a two-way transducer cannot do better on \( \pi \) than a one-way machine. Since \( \pi \) is prime in the 1WFT hierarchy \([\text{EKSW15}]\), either \( \pi \preceq_{\text{1WFT}} \alpha \) or \( \alpha \) is in the least 1WFT-degree, which is the set of ultimately periodic words (Remark 6.4).

Classifying all infinite strings may neither be relevant nor useful in practice. We now look at two particular classes of infinite words closed under 2WFT transformations: strings with decidable MSO-theory and computable strings. They describe subhierarchies whose study is initiated.
Proposition 6.8 (subhierarchies).

(i) If $\alpha \preceq_{2\text{WFT}} \beta$ and if $\beta$ is computable, then $\alpha$ is computable;

(ii) if $\alpha \preceq_{2\text{WFT}} \beta$ and if $\mathcal{M}^\beta$ has a decidable MSO-theory, then $\mathcal{M}^\alpha$ has a decidable MSO-theory.

Proof. (i) is immediate and (ii) follows from the equivalence with MSOT. \hfill \Box

Fact 6.9. The string $\pi$ has a decidable MSO-theory (see e.g. [Bár08]).

Proposition 6.10 ([EKSW15]). There exists a string $\tau$ such that for any computable string $\alpha$, $\alpha \preceq_{1\text{WFT}} \tau$.

Corollary 6.11. There exists a greatest degree of computable strings in the $2\text{WFT}$ hierarchy. It is the $2\text{WFT}$-degree of $\tau$.

Proof. For every computable string, we have $\alpha \preceq_{1\text{WFT}} \tau$, thus $\alpha \preceq_{2\text{WFT}} \tau$. \hfill \Box

Fact 6.12. The MSO theory of $\tau$ is not decidable, since there exists computable strings with undecidable MSO-theory.

Note that the $2\text{WFT}$-degree of ultimately periodic sequences, the $2\text{WFT}$-degree of $\pi$ and the $2\text{WFT}$-degree of $\tau$ must be distinct. Figure 2 summarizes the main ideas of this section.

Several challenging issues naturally arise around the structure of the $2\text{WFT}$ hierarchy and its subhierarchies. However, this is possibly a tough subject, since it is already quite hard to explicit the one-way case [EKSW15]. Among others, an interesting question is to describe the degrees of well-known sequences with decidable MSO-theory, such as morphic or $k$-morphic words [Bar08].
7 Conclusion and outlook

Preorders of advices, logic and transductions. Our first concern in this paper was the study various preorders over infinite words, related to the notion of advice strings. The results draw a generic correspondence between definability with advice, logical transductions and machine transductions. Table 1 summarizes this philosophy in an elegant way (the notion of (relativized) MSO-relabelings is less standard than MSOT). The gap between MSO-relabelings and MSOT shows that having basic knowledge on the languages is far from being sufficient to understand the richness of presentable structures.

| Advice  | $\text{Reg}$ | $\text{Reg}^\omega$ | $\text{AutStr}$ | $\text{AutStr}^\omega$ |
|---------|---------------|-----------------------|------------------|-----------------------|
| Logic   | rel. MSO-relabelings | MSO-relabelings          | MSOT             |
| Machine | Mealy machines   | $\omega$-regular functions | 2WFT             |

Table 1: Equivalent definitions for preorders over $\omega$-words

The two-way transductions hierarchy. Two-way transductions appear here more basic than relations defined by one-way machines, since they are clearly motivated by model-theoretical notions. Furthermore, it fits our informal conditions to be a “good” complexity measure over infinite words. A more involved study of the 2WFT hierarchy may help classifying certain hierarchies of structures, or even understand standard automatic presentations. We recall that such transductions over infinite words are (rather) unexplored.

Acknowledgements. This work would never have been possible without the support of Erich Grädel. I also thank Wolfgang Thomas, Christof Löding, Frédéric Reinhardt and anonymous referees for their advice.
References

[Abu16] Faried Abu Zaid. *Algorithmic solutions via model theoretic interpretations*. PhD thesis, 2016.

[AFT12] Rajeev Alur, Emmanuel Filiot, and Ashutosh Trivedi. Regular transformations of infinite strings. In *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, pages 65–74. IEEE Computer Society, 2012.

[AGR17] Faried Abu Zaid, Erich Grädel, and Frederic Reinhardt. Advice Automatic Structures and Uniformly Automatic Classes. In *26th EACSL Annual Conference on Computer Science Logic (CSL 2017)*, 2017.

[AS03] Jean-Paul Allouche and Jeffrey Shallit. *Automatic sequences: theory, applications, generalizations*. Cambridge university press, 2003.

[Bár08] Vince Bárány. A hierarchy of automatic ω-words having a decidable MSO theory. *RAIRO-Theoretical Informatics and Applications*, 42(3):417–450, 2008.

[Bel08] Aleksandrs Belovs. Some algebraic properties of machine poset of infinite words. *RAIRO-Theoretical Informatics and Applications*, 42(3):451–466, 2008.

[BG00] Achim Blumensath and Erich Grädel. Automatic structures. In *Logic in Computer Science, 2000. Proceedings. 15th Annual IEEE Symposium on*, pages 51–62. IEEE, 2000.

[BS69] Robert M. Baer and Edwin H. Spanier. Referenced automata and metaregular families. *Journal of Computer and System Sciences*, 3(4):423–446, 1969.

[CL12] Arnaud Carayol and Christof Löding. Uniformization in automata theory. In *International Congress of Logic, Methodology and Philosophy of Science. College Publications, London*, 2012.

[EH01] Joost Engelfriet and Hendrik Jan Hoogeboom. MSO definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic (TOCL)*, 2(2):216–254, 2001.

[EKSW15] Jörg Endrullis, Jan Willem Klop, Aleksi Saarela, and Markus Whitehead. Degrees of transducibility. In *International Conference on Combinatorics on Words*, pages 1–13. Springer, 2015.

[KN95] Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In *Logic and computational complexity*, pages 367–392. Springer, 1995.

[KNRS04] Bakhadyr Khoussainov, André Nies, Sasha Rubin, and Frank Stephan. Automatic structures: richness and limitations. In *Logic in Computer Science, 2004. Proceedings of the 19th Annual IEEE Symposium on*, pages 44–53. IEEE, 2004.
[KRSZ12] Alex Kruckman, Sasha Rubin, John Sheridan, and Ben Zax. A Myhill-Nerode theorem for automata with advice. In *Gandalf*, pages 238–246, 2012.

[LC07] Christof Löding and Thomas Colcombet. Transforming structures by set interpretations. *Logical Methods in Computer Science*, 3, 2007.

[Mil78] Terrence S Millar. Foundations of recursive model theory. *Annals of Mathematical Logic*, 13(1):45–72, 1978.

[OT05] Graham P Oliver and Richard M Thomas. Automatic presentations for finitely generated groups. In *STACS*, volume 3404, pages 693–704. Springer, 2005.

[Rab14] Alexander Rabinovich. A proof of kamp’s theorem. *Logical Methods in Computer Science*, 10(1), 2014.

[Rei13] Frédéric Reinhardt. Automatic structures with parameters. 2013.

[RT06] Alexander Rabinovich and Wolfgang Thomas. Decidable theories of the ordering of natural numbers with unary predicates. In *International Workshop on Computer Science Logic*, pages 562–574. Springer, 2006.

[Sac63] Gerald E Sacks. *Degrees of unsolvability*. Number 55. Princeton University Press, 1963.

[Sal68] Arto Salomaa. On finite automata with a time-variant structure. *Information and Control*, 13(2):85–98, 1968.

[Sec67] Dana Scott. Some definitional suggestions for automata theory. *Journal of Computer and System Sciences*, 1(2):187–212, 1967.

[Sem84] Aleksei Lvovich Semenov. Logical theories of one-place functions on the set of natural numbers. *Mathematics of the USSR-Izvestiya*, 22(3):587, 1984.

[Tho97] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of formal languages*, pages 389–455. Springer, 1997.

[Tsa11] Todor Tsankov. The additive group of the rationals does not have an automatic presentation. *The Journal of Symbolic Logic*, 76(04):1341–1351, 2011.
# Contents

1 Introduction

2 Preliminaries
   2.1 Words and languages ........................................ 3
   2.2 Structures and logic ....................................... 3
   2.3 Interpretations ............................................ 4

3 Simple case: regular languages with advice
   3.1 Terminating languages ................................. 6
   3.2 Non-terminating languages and $\omega$-regularity ........ 8

4 Advice automatic structures
   4.1 Tools and basic properties of advice presentations ... 10
   4.2 Terminating and non-terminating encodings ............. 11
   4.3 Digression: relativization of MSO-formulas ............ 13

5 Transductions and advice presentations
   5.1 MSO-transductions ...................................... 14
   5.2 From automatic structures to logical transductions ... 15
   5.3 An equivalent computational model ...................... 17

6 The two-way transductions hierarchy

7 Conclusion and outlook

A Proof of Proposition 3.18  .................................. 26

B Proof of Theorem 4.14 ....................................... 26
   B.1 FO-definable languages ................................. 26
   B.2 General case of $\omega$-regular languages ............. 27

C Self-contained proof of Corollary 5.9  .................... 29
   C.1 Homogeneous presentations ............................ 29
   C.2 Proof of Lemma C.3 ............................. 29
     C.2.1 A bounded-to-one index function ................. 30
     C.2.2 Back to homogeneous presentations ............. 31

D Proof of Theorem 5.21 ...................................... 32
   D.1 Streaming string transducers .......................... 32
   D.2 Transforming simple SST into 2WFT ................... 34

E Proof of Theorem 6.7 ....................................... 37
A Proof of Proposition 3.18

The proof detailed below uses the following lemma, that translates advice properties of unary languages into logical interpretations.

Lemma A.1. Let $\beta \in \Delta^\omega$. A language $U \subseteq 0^+$ is in $\text{Reg}^{\omega}[\beta]$ if (and only if) there is an $\text{MSO}[<, \Delta]$-formula $\phi(x)$ such that $\beta \models \phi(n) \iff 0^{n+1} \in U$.

Proof. Assume $U \in \text{Reg}^{\omega}[\beta]$. There is an $\omega$-regular language $L \subseteq (\{0, \Box\} \times \Delta)^\omega$ such that $U = \{ w \in 0^+ \mid w \otimes \alpha \in L \}$. Automata-logic translations show that the set $L$ can be described by an $\text{MSO}[<, \{0, \Box\} \times \Delta]$-sentence $\psi$. Let $\phi_n(x)$ be the $\text{MSO}[<, \Delta]$-formula obtained from $\psi$ by adding one free first-order variable $x$ and replacing each $P_{(0, \beta)}(y)$ by $P_0(y) \land y \leq x$ and each $P_{(\Box, \beta)}(y)$ by $P_0(y) \land y > x$. A simple induction shows that $\beta \models \phi(n) \iff 0^{n+1} \in U$.

The converse results from a similar argument.

\[ i) \Rightarrow iv) \] Assume that $\text{Reg}^{\omega}[\alpha] \subseteq \text{Reg}^{\omega}[\beta]$, then $\text{Pref}(\alpha) \in \text{Reg}^{\omega}[\beta]$. Closure properties of $\omega$-regular languages with advice (see Proposition 3.14) give that $L_a := \{0^{n+1} \mid |a[n]| = a \} \in \text{Reg}^{\omega}[\beta]$. Lemma A.1 then shows (iv).

\[ iv) \Rightarrow ii) \Rightarrow iii) \] If we have $(\phi_n(x))_{a \in \Gamma}$, let $\Phi$ be the formula $\bigwedge_{a \in \Gamma} (P_a(x) \leftrightarrow \phi_n(x))$ translated on the signature $\Gamma \times \Delta$. $\Phi$ describes an $\omega$-regular language $L' \subseteq (\Gamma \times \Delta)^\omega$ such that $\{a\} = \{ w \mid w \otimes \beta \in L' \}$. Thus $\{\alpha\} \in \omega\text{Reg}[\beta]$ and (ii) follows from closure properties of $\omega$-regular languages. On the other hand, uniformization theorems (transformations of relations into functions, see CL12) applied to $L'$ give (iii).

\[ (iii) \Rightarrow i) \] If $\alpha$ is the image of $\beta$ under some $\omega$-regular function, closure properties of $\omega$-regular languages show that every non-terminating automaton using $\alpha$ as an advice can also use $\beta$.

Remark A.2. The proof unsurprisingly gives the equivalence of inclusions with $\text{Pref}(\alpha) \in \text{Reg}^{\omega}[\beta]$ and $\{\alpha\} \in \omega\text{Reg}[\beta]$. These languages are, in a certain sense, complete for the advice classes.

B Proof of Theorem 4.14

We split the proof in two independent parts to treat either $\text{FO}$-definable languages only, or $\omega$-regular in a general way. The first case uses logical notions, whereas the second is highly based on structural properties of ($\omega$-)automata.

B.1 FO-definable languages

At first glance, there is no reason why we could only check a finite part of a suffix to deduce an $\text{FO}$-definable $\omega$-property. Indeed, it seems hard to formalise the intuition that, since our models all are suffixes of a given word, what happens infinitely often is always true, because first-order sentences are somehow too complex. However, there is no need to despair, because $\text{FO}$ has the same expressive power (over infinite words) as linear temporal logic (LTL) - a miraculous result known as Kamp’s theorem. In order to avoid possible confusions since several equivalent syntaxes exist, we now recall a syntax of LTL.
Definition B.1 (LTL[Γ]). The set of LTL[Γ]-formulas is defined inductively: every $a \in \Gamma$ is a formula, and $\top$ as well (atoms), if $\phi_1$ and $\phi_2$ are formulas, so are $\phi_1 \land \phi_2$, $\phi_1 \lor \phi_2$, $\neg \phi_1$, $X\phi_1$ and $\phi_1 U \phi_2$.

We may use the following abbreviations: $\bot := \neg \top$ and $G\phi := \neg(\top U \neg \phi)$. The semantic of LTL in $\omega$-words being well-known and quite intuitive, we do not recall it here.

Theorem B.2 (Kamp, future-only fragment \[Rab14\]). Over the word structures of $\Gamma^\omega$, for every FO[$<$, $\Gamma$]-sentence $\phi$, there is an equivalent $\phi' \in \text{LTL}[\Gamma]$.

The next step is to get a negation normal form.

Lemma B.3. For every LTL-formula, there is an equivalent formula where negations only appear in front of atoms, and other connectives are $U$, $G$, $X$, $\land$ and $\lor$.

Proof idea (folklore). Apply inductively the standard LTL equivalences, namely $\neg X\phi \equiv X\neg\phi$ and $\neg(\phi U \psi) \equiv (G \neg \psi) \lor (\neg \psi U (\neg \psi \land \neg \phi))$.

We fix an FO[$<$, $\Gamma$]-sentence $\phi$ and an infinite word $\alpha \in \Gamma^\omega$. By Theorem B.2 and Lemma B.3 there exists a LTL[Γ]-formula $\phi'$ in negation normal form and equivalent to $\phi$ (over all word models). The idea is now to remark that, since the word $\alpha$ is fixed, the connective $G$ is somehow useless. Let the $G$-subformulas of $\phi'$ be the largest subformulas whose main connective is $G$. If there is $n \geq 0$ such that $\alpha[n:] \models G \nu$, the $G$-subformula $G \nu$ is said to be consistent, in that case $\alpha[m:] \models G \nu$ for all $m \geq n$.

Let $\phi''$ be the formula obtained from $\phi'$ by replacing each consistent $G$-subformula by $\bot$ and each non-consistent $G$-subformula by $\bot$. $\phi''$ is in negation normal form and has no longer $G$ connectives. Furthermore, there is $N \geq 0$ such that for all $m \geq N$, $\alpha[m:] \models \phi''$ if and only if $\alpha[m:] \models \phi'$ (take $N$ to be the maximum of all the $n$ from the previous paragraph).

Now we translate back the formula $\phi''$ into an FO-sentence on finite words.

Lemma B.4. For every LTL[Γ]-formula $\phi''$ where negations only appear in front of atoms, and other connectives are $U, X, \land$ and $\lor$, there is a FO-sentence $\psi$ such that for all $\beta \in \Gamma^\omega$, $\beta \models \phi''$ if and only if there is $n \geq 0$ such that $\beta[n:] \models \psi$.

Proof idea. Induction on the LTL[Γ]-formula, all cases are trivial.

Since Lemma B.4 holds in particular for suffixes $\alpha[m:]$ of $\alpha$, we get that for all $m \geq N$, $\alpha[m:] \models \phi$ if and only if $\exists n \geq m, \alpha[m:n] \models \psi$.

B.2 General case of $\omega$-regular languages

We will follow a completely different scheme here. Our approach is based on results of A.L. Semenov \[Sem84\], where he gives a characterization of $\omega$-words whose MSO-theory is decidable. This main result is not useful in our context, but one of technical lemmas stated in the proof is of a particular relevance.

We briefly give here the definitions of \[Sem84\]. Recall that a congruence $\mathcal{E}$ on $\Gamma^\omega$ is an equivalence relation of finite index, compatible with concatenation. The key idea of Semenov lies in the following definition.
Definition B.5. Let $\mathcal{E}$ be a congruence, a $\mathcal{E}$-index $c$ is a nonempty finite word on the alphabet of equivalence classes, of length at most the index of the congruence. Given a $\mathcal{E}$-index $c$, define the set of its values $\text{val}(c) \subseteq \Gamma^*$ by induction:

- if $c$ is only one equivalence class $E$, then $\text{val}(c) = E \cap \Gamma$;
- if $c = c'E$ where $E$ is a class, then $w \in \text{val}(c)$ if and only three conditions are met: $w \in E$, every proper suffix/prefix of $w$ belongs to $\text{val}(c')$, every subword of $w$ belonging to $\text{val}(c')$ is either a suffix or a prefix.

It is not hard to see that for every $\mathcal{E}$-index $c$, $\text{val}(c)$ is a regular set which does not contain a proper subword of its words. The intuition behind this definition is that it helps computing the possible segments of runs in a given automaton (the congruence being based on the transition relation).

Denote by $\text{Index}(\mathcal{E})$ the finite set of all $\mathcal{E}$-indices. To quantify the occurrences of their values in a given $\alpha \in \Gamma^*$, let $\mathcal{E}^n\alpha := \{c \in \text{Index}(\mathcal{E}) \mid \exists w \in \text{val}(c)\text{subword of }\alpha[n:]\}$. It is clear that $\mathcal{E}^{n+1}\alpha \subseteq \mathcal{E}^n\alpha$, and since there is a finite number of indices, the sequence $(\mathcal{E}^n\alpha)_{n \geq 1}$ is ultimately constant with value $\mathcal{E}\alpha := \bigsqcup_{n \geq 1} \mathcal{E}^n\alpha$. A position $n \geq 0$ in a given $\alpha$ is said to be $\mathcal{E}$-remote if $\mathcal{E}^n\alpha = \mathcal{E}\alpha$ (we reached some kind of stability with respect to the congruence). Note that $\mathcal{E}$-remote positions of $\alpha$ form a final non-empty segment of $\mathbb{N}$.

Definition B.6 (regular system). Given a congruence $\mathcal{E}$, a regular $\mathcal{E}$-system $\mathcal{R}$ is a mapping from $\mathcal{P}(\text{Index}(\mathcal{E}))$ into regular subsets of $\Gamma^*$, such that the following property holds. For every $\omega$-word $\alpha$ and every position $n \geq 0$, there is a unique one position $m \geq n$ such that $\alpha[n : m] \in \mathcal{R}(\mathcal{E}\alpha)$.

Example B.7. Let $\mathcal{R}$ mapping any set $S$ of $\mathcal{E}$-indices to the set of words in $L := \bigcap_{\alpha \in S} \Gamma^* \text{val}(\alpha) \Gamma^*$ that have no proper prefix in $L$ (smallest elements for prefix-ordering). Then $\mathcal{R}$ is a $\mathcal{E}$-regular system.

If $n \geq 0$ is a position, denote by $\chi_n$ the characteristic sequence $0^n10^\omega$. We now state the result we need.

Lemma B.8 (Sem84, Lemma 8). For every $\omega$-regular language $L$, there is a congruence $\mathcal{E}$, a regular $\mathcal{E}$-system $\mathcal{R}$ and a finite word automaton $\mathcal{A}$ such that the following holds. For every $\alpha \in \Gamma^\omega$ and every $\mathcal{E}$-remote position $n$ (of $\alpha$), $\alpha \in L$ if and only if $\mathcal{A}$ accepts $\alpha[:m] \otimes \chi_n[:m]$ where $m \geq n$ is the unique position such that $\alpha[n:m] \in \mathcal{R}(\mathcal{E}\alpha)$.

This result is especially suitable in our context. Remark that given a congruence $\mathcal{E}$ and an infinite word $\alpha$, there is $N \geq 0$ such that for all $m \geq N$, every position is $\mathcal{E}$-remote in $\alpha[m:]$ (take $N$ to be any $\mathcal{E}$-remote position of $\alpha$). This allows us to reformulate a weaker version of Lemma B.8.

Lemma B.9. For every $\omega$-regular language $L$ and every infinite word $\alpha$, there exist $N \geq 0$ and a finite word automaton $\mathcal{D}$ such that for all $m \geq N$:

- if $\alpha[m:] \in L$ there is a (unique) $i_m \geq m$ such that $\mathcal{D}$ accepts $\alpha[m:i_m]$;
- else there is no $i \geq m$ such that $\mathcal{D}$ accepts $\alpha[m:i]$.

Proof. We use the notations of Lemma B.8. Since $\alpha$ is fixed, $R := \mathcal{R}(\mathcal{E}\alpha)$ is a fixed regular language verifying the subword property evoked in Definition B.6. Also note that for all $n \geq 0$, $\mathcal{R}(\mathcal{E}\alpha[n:]) = R$ since $\mathcal{E}\alpha[n:] = \mathcal{E}[\alpha]$. Choose $N$ to
be such that for all \( m \geq N \), every position in \( \alpha[m:] \) is \( \mathcal{E} \)-remote. In particular 0 is an \( \mathcal{E} \)-remote position of each \( \alpha[m:] \), hence we can get rid of the characteristic sequence \( \chi_n \) if we take this value. In other words, we build from Lemma 3.8 an automaton \( \mathcal{A}' \) such that for all \( m \geq N \), \( \alpha[m:] \in L \) if and only if \( \mathcal{A}' \) accepts \( \alpha[m : m'] \) where \( m' \geq m \) is the unique position such that \( \alpha[m : m'] \in R \).

Now, the construction of \( \mathcal{D} \) is based on the product of \( \mathcal{A}' \) and an automaton \( \mathcal{A}'' \) recognizing \( R \). It then checks if \( \mathcal{A}' \) accepts when \( \mathcal{A}'' \) accepts, what necessarily happens after a finite time.

**Lemma 3.9** provides exactly the elements we needed to achieve the proof.

### C Self-contained proof of Corollary 5.9

We show that \( P_f(W\alpha) \in \text{AutStr}_\infty[\beta] \) implies \( \alpha \preceq_{\text{MSOT}} \beta \). We actually adapt and shorten the generic proof of [LC07] in our context.

#### C.1 Homogeneous presentations

We first present a simple class of \( \text{Reg}_\infty[\beta] \)-presentation, that generalize presentations over a unary alphabet.

**Definition C.1** (homogeneity). A \( \text{Reg}_\infty[\beta] \)-presentation is said to be homogeneous if there is an integer \( K \geq 1 \) such that the finite words encoding the elements belong to \( \bigcup_{1 \leq k \leq K} k^* \).

Such presentations are especially simple, in the sense that the only relevant information in a word is its length (and its “color” \( k \), which is bounded). They are closely related to MSO-transductions, as shown in Lemma C.2.

**Lemma C.2.** If \( W\alpha \) has an homogeneous \( \text{Reg}_\infty[\beta] \)-presentation, \( \alpha \preceq_{\text{MSOT}} \beta \).

**Proof sketch.** Let \( L \subseteq \bigcup_{1 \leq k \leq K} k^* \) be the language encoding the domain of \( W\alpha \) in the presentation. We build a \( K \)-copying MSO-transduction \( \tau \) such that \( \tau(\beta) = \alpha \). Indeed, according to Lemma A.1 for \( 1 \leq k \leq K \) fixed, the set of positions \( \{ n \mid k^n \in L \} \) can be described by a formula \( \phi_k(x) \). A similar argument show that there exists a formula \( \phi_<(x,y) \) to describe the relation \(<\).

**Lemma C.3.** If \( P_f(\mathfrak{A}) \in \text{AutStr}_\infty[\beta] \), then \( \mathfrak{A} \) has an homogeneous \( \text{Reg}_\infty[\beta] \)-presentation.

Assume Lemma C.3 holds. If \( P_f(\mathfrak{A}) \in \text{AutStr}_\infty[\beta] \) then \( W\alpha \) has an homogeneous \( \text{Reg}_\infty[\beta] \)-presentation, thus \( \alpha \preceq_{\text{MSOT}} \beta \) by Lemma C.2.

The rest of this section is devoted to the proof of Lemma C.3.

#### C.2 Proof of Lemma C.3

Let \( \mathfrak{A} \) be a structure of domain \( A \) such that \( P_f(\mathfrak{A}) \in \text{AutStr}_\infty[\beta] \). The \( \text{Reg}_\infty[\beta] \)-presentation of \( P_f(\mathfrak{A}) \) can be assumed injective by Proposition 4.9. Let \( \Sigma \) the encoding alphabet and \( \nu : P_f(A) \rightarrow \Sigma^* \) the encoding function.
Let Atoms := \{w \in \Sigma^\omega \mid \nu(w) \text{ is a singleton}\}. Our main purpose is to number the elements of Atoms in a regular-like way. More formally, we build a function 

\text{Index} : \text{Atoms} \rightarrow \mathbb{N}\text{ such that } |\text{Index}^{-1}(n)| \leq K \text{ for all } n \geq 0 \text{ and the language } \{w \otimes 0^n \in \Sigma^\omega \mid \text{Index}(w) = n\} \text{ is } \omega\text{-regular with advice } \beta (K \text{ being a large enough constant). Once this is done, the lemma follows almost directly. We denote by } L_\subseteq \text{ the language } \{w \otimes w' \in \Sigma^\omega \mid \nu(w) \subseteq \nu(w')\}.

C.2.1 A bounded-to-one index function

Thanks to the \text{Reg}^\infty[\beta]\text{-presentation}, there exist two (deterministic Müller) \omega-automata \mathcal{A}_\text{Atoms} and \mathcal{A}_\subseteq such that Atoms = \{w \mid w \otimes \beta \in \mathcal{L}(\mathcal{A}_\text{Atoms})\} and 

L_\subseteq = \{w \mid w \otimes \beta \in \mathcal{L}(\mathcal{A}_\subseteq)\} \text{ (recall that here } w \text{ itself is a convolution). Let } 

Q_\text{Atoms} \text{ and } Q_\subseteq \text{ be the sets of states of these two automata and let the constant } K_{im} := (2|Q_\subseteq| + 1)|Q_\text{Atoms}| \geq 2.

Definition C.4. Let \( w \in \text{Atoms} \), the set of its important positions \text{Imp}(w) \subseteq \mathbb{N} is such that \( n \in \text{Imp}(w) \) if and only if \( |\{w' \in \Sigma^\omega \mid w[n]w' \in \text{Atoms}\}| > K_{im} \).

\text{Imp}(w) \text{ can be seen as the set of prefixes } w[n] \text{ whose lecture does not give much information on who is } w. \text{ Note that it is an initial segment of } \mathbb{N} \text{ (we can w.l.o.g. assume that it is always non-empty as soon as } A \text{ is not finite). Furthermore, if } w = v \otimes \omega, \text{ the elements } \text{Imp}(w) \text{ are smaller than } |v| + 1. \text{ Therefore the following definition makes sense. }

Definition C.5. If \( w \in \text{Atoms} \), \text{Index}(w) \in \mathbb{N} \text{ is max } \text{Imp}(w).

Before giving the essential property of \text{Index}, we first remark that \{w \otimes 0^n \in \Sigma^\omega \mid \text{Index}(w) = n\} \text{ is } \omega\text{-regular with advice } \beta \text{ (closure properties).}

Lemma C.6. There is a constant } K \text{ such that for all } n \geq 0, |\text{Index}^{-1}(n)| \leq K.

The rest of this subsubsection is dedicated to the combinatorial proof of this lemma. We first give some notations. Let \( \delta_\text{Atoms} \) be the transition function of \( \mathcal{A}_\text{Atoms} \) with advice \( \beta \) (for a finite word \( w \), \( \delta_\text{Atoms}(w) \) is the state reached after reading \( w \otimes \beta[; w]\)). Let \( L^n_{\text{Atoms}}(q) \) be the partial language of \( \mathcal{A}_\text{Atoms} \) in \( q \) with advice \( \beta[n;] \) (infinite words accepted starting in the state \( q \) with advice \( \beta[n;]\)).

We use similar notations \( \delta_\subseteq \) and \( L^n_\subseteq \) for the automaton \( \mathcal{A}_\subseteq \).

Lemma C.7. Let \( K_1 := 2|Q_\subseteq| + 1 \). For all \( n \geq 0 \) and every state \( q \in Q_\text{Atoms} \), either \( |L^{n+1}_\text{Atoms}(q)| < K_1 \text{ or } |\{v \mid |v| = n \text{ and } \delta_\text{Atoms}(v) = q\}| < K_1 \).

Proof. Let \( B := \{v \mid |v| = n \text{ and } \delta_\text{Atoms}(v) = q\} \).

Assume that you have \( K_1 \) distincts (finite) words \( v_1 \ldots v_{K_1} \) in \( B \), and \( K_1 \) distincts (infinite) words \( v_1' \ldots v_{K_1}' \) in \( L^n_{\text{Atoms}}(q) \). Define then \( w_{i,j} := v_i v'_j \) for \( 1 \leq i, j \leq K_1 \). According to the definitions, we have \( w_{i,j} \in \text{Atoms} \) (since \( v_i \) leads to \( q \) in \( n \) steps and \( v'_j \) is accepted starting in \( q \) at time \( n \)).

Let \( W := \{w_{i,j} \mid 1 \leq i, j \leq K_1\} \). Let \( W' \subseteq \text{dom}(S) \) be the set of elements encoded (as singletons) by the words of \( W' \). Then \( |W'| = (K_1)^2 \). Let \( P = \mathcal{P}(W') \subseteq \mathcal{P}(\text{dom}(S)) \) and \( C \) the set of (infinite) words encoding the elements of \( P \) in the presentation of \( \mathcal{P}(S) \). Then \( |C| = |P| = 2^{(K_1)^2} \).

For each \( w \in C \) define:

- \( d_w : [1, K_1] \rightarrow Q_\subseteq \) mapping \( i \) to \( \delta_\subseteq(v_i \otimes w[; n]) \);
Atoms

Contradiction.

Elements in distincts such that for each one \( n \) and \( \delta \).

Indeed, this means that exactly the same \( v_{i,j} \) are in relation \( w_1 \) and \( w_2 \).

Now there are \( \left| Q_{\subseteq} \right|^F = 2^{2^{|Q_{\subseteq}||K_1}} \leq 2^{|Q_{\subseteq}||K_1} \) different \( d_w \), and \( 2^{|Q_{\subseteq}||K_1} \) different \( f_w \) possible. This meaning that we can define at most \( 2^{|Q_{\subseteq}||K_1} \) elements in \( C \), so we have \( (K_1)^2 \leq 2^{|Q_{\subseteq}||K_1} = (K_1 - 1)K_1 \) what brings a contradiction.

Now we can forget everything about the powerset and work only with the atoms and Lemma C.7. The following result shows that the number of possible prefixes before the index is bounded.

**Lemma C.8.** Let \( I_n = \{ w[j] \mid w \in \text{Atoms and } \text{Index}(w) = n \} \), then all \( n \geq 0 \), \( |I_n| < K_{im} \).

**Proof.** Assume that \( |I_n| \geq K_{im} \), then remark that \( K_{im} = |Q_{\text{Atoms}}|K_1 \). Our assumption implies that there is a state \( q \in Q_{\text{Atoms}} \) and \( v_1 \ldots v_{K_1} \in I_n \) distincts such that for each one \( \delta_{\text{Atoms}}(v_i) = q \). This implies that \( |v| = n \) and \( \delta_{\text{Atoms}}(v) = q \) \( \geq K_1 \), therefore by Lemma C.7 we have \( |L_{\text{Atoms}}^n(q)| < K_1 \).

Note that there must be (at least) an infinite word \( v_1 v' \in \text{Atoms} \) whose index is \( n \) (by definition of \( v_1 \)). In particular, \( n \) is an important position and \( \{ w \mid v_1 w \in \text{Atoms} \} = L_{\text{Atoms}}^n(q) \) has more than \( K_{im} \geq K_1 \) elements. **Contradiction.**

Next, we show that the number of possible suffices after the index is bounded.

**Lemma C.9.** Let \( J_n = \{ w[n] \mid w \in \text{Atoms and } \text{Index}(w) = n \} \), then for all \( n \geq 0 \) we have \( |J_n| \leq |Q_{\text{Atoms}}| |\Sigma| \).

**Proof.** Since \( |J_n| \leq |\Sigma| |B| \), where \( B := \{ w[n+1] \mid w \in \text{Atoms and } \text{Index}(w) = n \} \), we only need to bound the size of \( B \) by \( K_{im} |Q_{\text{Atoms}}| \).

For all \( v' \in B \), there exist a state \( q \in Q_{\text{Atoms}} \) and a word \( v \) of size \( n+1 \) such that \( \delta_{\text{Atoms}}(v) = q \), \( v' \in L_{\text{Atoms}}^n(q) \) and \( \text{Index}(v v') = n \) (by definition). In particular, this means that \( n + 1 \notin \text{Imp}(v v') \). By definition of Imp, \( \{ v, v v' \in \text{Atoms} \} \leq K_{im} \). But \( L_{\text{Atoms}}^{n+1}(q) \leq \{ v, v v' \in \text{Atoms} \} \). Therefore \( |L_{\text{Atoms}}^{n+1}(q)| \leq K_{im} \).

Since each \( v' \in B \) comes from (at least) one such \( L_{\text{Atoms}}^{n+1}(q) \) (for a certain \( q \in Q_{\text{Atoms}} \)), we have shown \( |B| \leq K_{im} |Q_{\text{Atoms}}| \).

**Lemma C.6** follows directly from Lemmas C.8 et C.9. Indeed, if \( \text{Index}(w) = n \), then \( [n] \in I_n \) and \( w[n] \in J_n \), this meaning that \( |\text{Index}^{-1}(n)| \leq K := K_{im} \times (K_{im} |Q_{\text{Atoms}}| |\Sigma|) \).

**C.2.2 Back to homogeneous presentations**

We come back to the proof of Lemma C.8. Recall that we want to show that \( \mathfrak{A} \) has an homogeneous \( \text{Reg}^\infty[\beta] \)-presentation. We have built a function Index : \( \text{Atoms} \rightarrow \mathfrak{N} \) with the following properties:

- \( \{ w \otimes 0^n \square w' \mid \text{Index}(w) = n \} \in \omega \text{Reg}[\beta] \);
there is $K$ such that $|\text{Index}^{-1}(n)| \leq K$ for all $n$.

There exists a well-ordering over finite words whose graph is regular. Hence if $1 \ldots K$ are new letters, we can order the elements of each $\text{Index}^{-1}(n)$ such that $\{w \otimes k^n \sqcup \omega \mid w$ is the $k$-th word in $\text{Index}^{-1}(n)\} \in \omega \text{Reg}[\beta]$.

This ordering defines an injective function $\text{Code} : \text{Atoms} \to \bigcup_{1 \leq k \leq K} k^*$ mapping $w$ to the unique $k^n$ such that $w \otimes k^n \sqcup \omega \in L$. Furthermore the language $L := \{k^n \sqcup \omega \mid \exists w \text{Code}(w) = k^n\}$ is in $\omega \text{Reg}[\beta]$. The function $\eta : L \to A$ defined by $\eta(k^n \sqcup \omega) = \nu(\text{Code}^{-1}(k^n))$ and is a bijection between $L$ and $A$.

It is then easy to check that $\eta$ is the encoding function of an (injective) $\omega \text{Reg}^{\infty}[\beta]$-presentation of $A$ over the language $L$. Since all the words of $L$ are of the form $k^n \sqcup \omega$, removing the padding symbols $\sqcup$ from the encodings provides an homogeneous $\text{Reg}^{\infty}[\beta]$-presentation.

**D Proof of Theorem 5.21**

As evoked in the main body of this paper, we only show that $\alpha \preceq_{\text{MSOT}} \beta$ implies $\alpha \preceq_{2\text{WFT}} \beta$. The sketch of this proof is the following. We first recall a result from [AFT12]: if $\alpha \preceq_{\text{MSOT}} \beta$, then $\alpha$ can be computed from $\beta$ by a streaming string transducer. Our work lies in the transformation of such a transducer into a $2\text{WFT}$ when fixing its input. This result is not the same as Theorem 5.20, since we do not want to obtain $\omega$-lookaheads in the resulting transducer.

**D.1 Streaming string transducers**

Informally, a *streaming string transducer* is a one-way transducer with a finite set $X$ of registers that can store information and be updated in a simple way. We call substitution a mapping from $X$ to $(\Gamma \sqcup X)^*$. A substitution $\sigma$ is said to be copyless if each variable appears at most once in the hole set $\{\sigma(x) \mid x \in X\}$. Let $S_X$ be the set of all copyless substitutions. Substitutions can be extended from $(\Gamma \sqcup X)^*$ to $(\Gamma \sqcup X)^*$ and thus composed.

**Example D.1.** $x \mapsto x, y \mapsto x$ is not copyless, but $x \mapsto z, y \mapsto yx$ is so.

**Definition D.2 ([AFT12]).** A (deterministic) streaming $\omega$-string transducer (SST) is an 8-tuple $S = (Q, q_0, \Delta, \Gamma, \delta, X, \lambda, F)$ where:

- $Q$ is the finite set of states with initial state $q_0$;
- $\Delta$ (resp. $\Gamma$) is the input (resp. output) alphabet;
- $\delta : Q \times \Delta \to Q$ is the (partial) transition function;
- $X$ is a finite set of registers (also called variables);
- $\lambda : Q \times \Delta \to$ is the (partial) register update;
- $F : \mathcal{P}(Q) \to X^*$ is the (partial) output function such that for $P \in \text{dom}(F)$ the string $F(P)$ is copyless of the form $x_1 \cdots x_n$ and for all $q, q' \in P$ with $q' = \delta(q, a)$ we have $\rho(q, a)(x_i) = x_i$ for all $i < n$ and $\lambda(q, a)(x_n) = x_n u$ for some $u \in \Gamma \cup X^*$.
If $P$ is the set of states that appear infinitely often in the run of $S$ on $\beta \in \Delta^\omega$, we want the value of $F(P) = x_1 \cdots x_n$ to be the (infinite) output string. Thus we need this value to change in a convergent way, so we force $\lambda(q,a)(x_i) = x_i$ and $\lambda(q,a)(x_n) = x_n$ when the transition remains in $P$. The modification only adds something in the end of the value of $x_1 \ldots x_n$.

More formally, a run $\rho$ of $S$ on $\beta \in \Delta^\omega$ is a run of the underlying deterministic automaton $(Q,q_0,\Delta,\delta)$: for $k \geq 0$, $\rho(k)$ is the state reached after reading $\alpha[n]$. We define inductively a sequence of ground (from $X$ to $\Gamma^*$) substitutions $(\sigma_k)_{k \geq 0}$: $\sigma_0(x) = \varepsilon$ for all $x$ and $\sigma_{k+1} = \sigma_k \circ \lambda(\rho(k),\alpha[k])$. If $P$ is the set of states appearing infinitely often in $\rho$ and $F(P) = x_1 \cdots x_n$, then necessarily $\lim_k \sigma_k(x_1 \cdots x_n)$ converges to a (finite or infinite) word. If finite, we make it infinite adding $\square \omega$ in the end. It defines the output of $S$ on $\beta$.

**Example D.3.** In Figure 3 we give an SST that outputs the word $\tilde{\alpha}$ on the input $\alpha \in (\Delta^\omega \setminus \#)^\omega$. Let $X := \{x, \text{out}\}$, and $F(\{q_0\}) = \text{out}$; transitions are represented with the notation [input | substitution] and $\alpha$ stands for every symbol except $\#$.

$$
\begin{array}{c|cc}
\alpha & x \rightarrow ax, & \text{out} \rightarrow \text{out} \\
\# & x \rightarrow \varepsilon, & \text{out} \rightarrow \text{out} x \\
\end{array}
$$

Figure 3: A simple SST

**Theorem D.4 ([AFT12], Theorem 2).** SST-definable functions between infinite strings are exactly MSOT-definable functions.

We shall only need a weaker reformulation of this result, i.e. if $\alpha \equiv_{\text{MSOT}} \beta$ then $\alpha$ is computable from $\beta$ by an SST. The model can even be simplified in that case, as shown in Corollary D.7.

**Definition D.5.** A simple SST is an SST with a distinguished register $\text{out} \in X$ such that $\text{dom}(F) = 2^Q$, and for all set of states $P$, $F(P) = \text{out}$.

**Example D.6.** The SST of Example D.3 is simple.

**Corollary D.7.** If $\alpha \equiv_{\text{MSOT}} \beta$ then $\alpha$ is the image of $\beta$ under some function realized by a simple SST.

**Proof.** By applying Theorem D.4 (in its weaker reformulation), if $\alpha \equiv_{\text{MSOT}} \beta$ then $\alpha$ is the image of $\beta$ under some SST $S := (Q,q_0,\Delta,\delta,X,\lambda,F)$. Let $\rho$ be the run on $\beta$, $(\sigma_k)_{k \geq 2}$ the associated sequence of ground substitutions, $P$ be the set of states that appear infinitely often in $\rho$. Necessarily $P \in \text{dom}(F)$ thus $F(P) = x_1 \cdots x_n$. This sequence of registers was unpredictable when the input could change, but now it is fixed since $\beta$ is so. Let $K \geq 0$ such that for all $k \geq K$, $\rho(k) \in P$. For $x \in X$ let $w_x := \sigma_K(x) \in \Gamma^*$. According to the remarks above, if $i < n$ then for all $k \geq K$, $\sigma_k(x_i) = w_{x_i}$.

We define a simple SST $S'$. The set of registers is $X \uplus \{\text{out}\}$ and the states are $\{0 \ldots K-1\} \uplus P$. The graph of this transducer begins with a line of length

33
\(K\) on \(\{0 \ldots K - 1\}\) with trivial updates of the registers, and in the last transition the value of \(x \in \mathcal{X}\) is updated to \(w_n\) and \(\text{out}\) is updated to \(w_{x_n} \ldots w_{x_0}\). This transition leads to the state \(p(K) \in P\) and then \(S'\) moves in \(P\) like \(S\) does. The updates of \(\text{out}\) are defined like the updates of \(x_n\). We define the output function of \(S'\) so that it fulfills the requirements of a simple SST. It is easy to check that \(\alpha\) is the image of \(\beta\) under the function realized by \(S'\). Note that this transducer has no reason to preserve the output of \(S\) on other words.

In the definition of a simple SST, we thus replace \(F\) by \(\text{out}\). Recall nevertheless that \(\lambda(p, a)(\text{out})\) (if defined) must be of the form \(\text{out}\ w\) with \(w \in (\Gamma \uplus \mathcal{X})^*\) and no other register can use the value of \(\text{out}\) in its update.

### D.2 Transforming simple SST into 2WFT

We first show how to transform a simple SST into a two-way transducer with a lookbehind feature. Informally, such a transducer has access to the state of a finite automaton reading a prefix of the input word.

**Definition D.8.** A two-way transducer with lookbehind (2WFT) is a two-way transducer \(T = (Q, q_0, \Delta \uplus \{\rhd\}, \Gamma, \delta, \theta)\) together with a deterministic automaton \(A = (S, s_0, \Delta, \zeta)\) (\(S\) set of states, \(s_0 \in S\) initial state, \(\zeta\) (partial) transition function), such that \(\delta : Q \times (\Delta \uplus \{\rhd\}) \times S \to Q \times \{\rhd, \lhd\}\) (partial function).

When in state \(q \in Q\) and position \(n\) of the input \(\beta \in \Delta^\omega\), the transition of the 2WFT is chosen as a function of \(q, \beta[n]\) and \(s := \zeta(s_0, \beta[:n])\). Note that the access to \(\zeta(s_0, \beta[:n])\) is purely an oracle and requires no effective run of \(A\). The definition of a run and the \(\text{out}\) output mechanism of a 2WFT is straightforward and similar to that of a 2WFT.

**Lemma D.9.** If \(\alpha\) is the image of \(\beta\) under a simple SST, then \(\alpha\) is the image of \(\beta\) under some 2WFT.

**Proof sketch.** Let \(S = (S, s_0, \Delta, \Gamma, \zeta, \mathcal{X}, \lambda, \text{out})\) be a simple SST and \(\beta \in \Delta^\omega\) such that \(\alpha \in \Gamma^\omega\) is output by \(S\) on input \(\beta\). We denote by \(\rho\) the corresponding run and \((\sigma^k)_{k \geq 0}\) the sequence of ground substitutions. We are going to build a 2WFT \(T\) outputting \(\alpha\) on \(\beta\). Let \(A = (S, s_0, \Delta, \zeta)\) be the deterministic automaton of \(S\), this automaton will be the lookahead of \(T\). In other words, we can assume that when in position \(k + 1\) of the input tape, \(T\) chooses its transition depending on \((\lhd \beta)[k + 1] = \beta[k]\) and \(\rho(k) = \zeta(s_0, \beta[:k])\). Hence the transition can also depend on \(\lambda(\rho(k), \beta[k])\).

Since an formal construction of \(T\) would needlessly burden this proof, we shall only provide a pseudocode describing its behavior. The main issue is that since \(T\) has no registers, it cannot store unbounded information before outputting, as \(S\) used to do. Hence, we shall use the two-way moves of \(T\) to compute “recursively” the value of the registers at each step, and output immediately what was added to \(\text{out}\). But the recursion procedure has to be quite subtle, since we have no memory to store an unbounded stack.

The code is given in Algorithm \(\mathcal{H}\) we now justify its correction, i.e. that it computes \(\alpha\) on input \(\beta\). Recall that the input tape contains in fact \(\lhd \beta\), hence the first “move right” sends the reading head on \(\beta[0]\).
Algorithm 1: the transducer $\mathcal{T}$

Variables. All variables are global. $\text{pos} \in \mathbb{N}$ denotes the current position on the input tape, implicitly updated at each move; $\text{reg} \in \mathcal{X}$ is the register we are currently working on; $\text{process} \in (\mathcal{X} \cup \Gamma)^*$ is what remains to be output for $\text{reg}$; $x \in \mathcal{X} \cup \Gamma$ will be used temporarily.

Function $\text{Next}$

\[
\begin{align*}
\text{if } & \text{process} \neq \varepsilon \text{ then} \\
& x \leftarrow \text{process}[0]; \\
& \text{process} \leftarrow \text{process}[1:] ; \\
& \text{if } x \in \mathcal{X} \text{ then} \\
& \quad \text{move left;} \\
& \quad \text{reg} \leftarrow x; \\
& \quad \text{if } (\vdash \beta)[\text{pos}] \vdash \text{ then} \\
& \quad \quad \text{process} \leftarrow \varepsilon; \\
& \quad \text{else} \\
& \quad \quad \text{process} \leftarrow \lambda(\rho(\text{pos} - 1), (\vdash \beta)[\text{pos]})(\text{reg}); \\
& \text{else} \\
& \quad \text{output } x; \\
& \text{end}
\end{align*}
\]

\[
\begin{align*}
& \text{else} \\
& \quad \text{if } \text{reg} = \text{out} \text{ then} \\
& \quad \quad \text{break the inner while of the main program;} \\
& \quad \text{else} \\
& \quad \quad \text{move right;} \\
& \quad \quad \text{find the unique } x \in \mathcal{X}, w_1 \in (\Gamma \cup \mathcal{X})^*, w_2 \in (\Gamma \cup \mathcal{X})^* \text{ such that} \\
& \quad \quad \lambda(\rho(\text{pos} - 1), (\vdash \beta)[\text{pos}]) (x) = w_1 \text{reg} w_2; \\
& \quad \quad \text{reg} \leftarrow x; \\
& \quad \quad \text{process} \leftarrow w_2; \\
& \text{end}
\end{align*}
\]

while true do

\[
\begin{align*}
& \text{move right;} \\
& \text{reg} \leftarrow \text{out;} \\
& \text{process} \leftarrow \lambda(\rho(\text{pos} - 1), (\vdash \beta)[\text{pos}])(\text{out}); \\
& \text{while true do} \\
& \quad \text{Next();} \\
& \text{end}
\end{align*}
\]

end
Algorithm 1 really describes a 2WFT. Variables only contain a bounded information, except pos but it is not directly used in the computations. Hence we describe a finite-memory machine. Conditions of the “if” depend on information which is available by a 2WFT.

Instruction “find the unique”. It is not clear that such $x, w_1, w_2$ exist and are unique. Their existence will be ensured at runtime. Uniqueness follows directly from the fact that the $\lambda$ substitutions are copyless. Note that this instruction is the key argument to make the procedure work without a stack of unbounded size.

Specifications of Next. The key invariant is the following. Let $k + 1$, $x$ and $w$ be the values of pos, reg and process at a certain instant, such that $k \geq 0$ and $w$ is a suffix of $\lambda(\rho(k), \beta[k])(x)$. We claim that after a certain number of calls to Next(), we have pos = $k + 1$, reg = $x$, process = $\epsilon$, and during this time $T$ has output $\sigma_k(w) \in \Gamma^*$. This result can be shown by induction on $(k, |w|)$ with lexicographical ordering.

Main invariant. Let $w_k \in (X \cup \Gamma)^*$ such that $\lambda(\rho(k), \beta[k])(\text{out}) = \text{out} \cdot w_k$. Using what was done for Next we get that after the $(k + 1)$-th main “while”, $T$ has output the string $\sigma_0(w_0) \cdots \sigma_k(w_k) \in \Gamma^*$. The previous invariant shows that $T$ outputs $\alpha$.

A run of this algorithm is detailed in Example D.10 below.

Example D.10. Let $X = \{x, y, \text{out}\}$ and $\Gamma = \{a, b\}$. The last substitutions applied are written under the positions $k - 1$, $k$ and $k + 1$. We want to output the last value of $xa$ added to out in position $k + 1$.

The 2WFT moves left from $(k + 1)$ to $k$ to find $\sigma_{k+1}(x)$ (since $x$ is the first variable appearing in $xa$). It can already output $b$, then goes left to look for $\sigma_k(y)$. It outputs a and reaches the end of a branch, so it moves right to $(k)$, keeping in memory that the last considered variable was $y$, which only appears as a right member in $x \mapsto \text{byx}$. So the next value to output is $\sigma_k(x)$, we do it with the same procedure. When it has finally output the full value of $\sigma_{k+1}(x)$, it moves right and notice that the whole recursion process ends after outputting $a$.

Remark D.11. We could in fact show that if a function is realizable by a simple SST, it can be realized by a 2WFT as well, but it is not useful in our context. This refined statement does not hold for SST in general.

The next lemma shows that adding lookbehinds does not increase the expressive power of the transducers.

Lemma D.12. If $\alpha$ is the image of $\beta$ under a 2WFT, then $\alpha$ is the image of $\beta$ under some 2WFT (without lookbehind).
Proof idea. \cite{EH01} shows how to remove lookbehinds in the case of 2WFT over finite words. Since the “behind” only concerns a finite part of our infinite string, the adaptation is straightforward.

We can now complete the proof of Theorem 5.21. If \( \alpha \leq_{\text{MSOT}} \beta \), then by Corollary D.7, \( \alpha \) is the image of \( \beta \) under a simple SST. By Lemma D.9, \( \alpha \) is the image of \( \beta \) under a 2WFT. Finally, Lemma D.12 concludes that \( \alpha \leq_{2\text{WFT}} \beta \).

E Proof of Theorem 6.7

We proceed in several steps to get that \( \alpha \leq_{2\text{WFT}} \pi \) implies \( \alpha \leq_{1\text{WFT}} \pi \). First (Lemma E.1), we show that a two-way computation on \( \pi \) can be performed by a transducer that only changes its reading direction when seeing the letter 1. We further prove (Lemma E.3) that it can be simulated by a one-way transduction in a “bigger” word, which is in the same 1WFT degree as \( \pi \) (Lemma E.4).

We assume there that \( \alpha \) is not ultimately periodic. According to Lemma 5.18, \( n \geq 0 \), there is a moment where the transducer no longer goes before position \( n \) in its input tape. Our constructions will refer implicitly to what happens “far enough” in the word.

**Lemma E.1.** If \( \alpha \leq_{2\text{WFT}} \pi \), the transformation can be performed by a transducer whose states \( Q \) are partitioned in two sets \( Q \ll \) and \( Q \gg \) such that the following holds for the transition function \( \delta \). For all \( q \in Q \ll \) (resp. \( Q \gg \)) \( \delta(q,0) = (q',\ll) \) (resp. \( (q',\gg) \); for all \( q \in Q \) and \( a \in \{0,1\} \), \( \delta(q,a) = (q',\ll) \) (resp. \( (q',\gg) \)) implies \( q' \in Q \ll \) (resp. \( q' \in Q \gg \)).

**Proof sketch.** Let \( T \) be a \( N \)-states transducer performing the transformation. We study how \( T \) copes with the \( 10^n1 \) blocks of \( \pi \). Assume \( T \) enters the block from the left side and goes right after reading the 1. We consider the (two-way) run staying in \( 0^n \), before the next visit of a letter 1. Since \( T \) can only see 0, it is finally (after a most \( N \) steps) caught in a loop (of size at most \( N \)). By loop we mean a two-way loop, between by two configurations sharing the same state. Two cases may occur.

- The next 1 visited is the left one: \( T \) comes back to its previous position. In that case, the run in \( 0^n \) cannot be longer than \( N + N^2 \) (due to the loop), else it should go “right”.

- The next 1 visited is the right one: \( T \) went through to block \( 0^n \).

For \( n \) large enough, the occurring case does not depends on \( n \), but only on the state when entering the block. It is not hard to derive a formal construction from the previous remark: the first case can be hardcoded without moving (since the run is bounded), and the second one can be simulated in a one-way manner (simulate a bounded loop). Adapt consequently the output function. We get a new transducer that does not change its reading direction in blocks of 0.

**Remark E.2.** This lemma remains valid when replacing \( \pi \) by any binary sequence where the gap between two consecutive 1 goes to the infinity.

Let \( \pi^k = \prod_{n=0}^{\infty} (0^n1)^k \) for \( k \geq 1 \).

**Lemma E.3.** \( \alpha \leq_{2\text{WFT}} \pi \), there is \( k \geq 1 \) such that \( \alpha \leq_{1\text{WFT}} \pi^k \).
Proof sketch. Let $\mathcal{T}$ be the $N$-states transducer obtained by Lemma [E.1]. Let $p_n$ for $n \geq 0$ be the position of the $(n+1)$-th letter 1 in $\pi$, namely $p_n = \frac{n(n+1)}{2}$. A non-trivial run on $\pi$ can be decomposed in finite runs $R_n$ between the last visit in $p_n$ and the last visit in $p_{n+1}$.

We claim that there is $c \geq 1$ such that for all $n$ large enough, $R_n$ never visits position $p_{n+c}$. Thus $R_n$ is contained between $p_n$ and $p_{n+c}$, so in $10^n1 \cdots 0^{n+c}1$. It follows that $R_n$ can be simulated by a one-way run on $(0^n)^cN$, by unfolding of the two-way moves (we use here the previous lemma). Putting everything together, we get that $\alpha \preceq_{1WFT} \pi^cN$.

Let us now prove what we claimed. The idea is that if $\mathcal{T}$ goes arbitrarily far in $R_n$, it must be caught in a (two-way) loop and thus cannot come back to $p_{n+1}$. This is not totally trivial, because the $10^n1$ blocks are of increasing size in the word, what might “break” the loop. The proof can be saved anyway. Let $\mathcal{A}$ be the underlying (deterministic) automaton of $\mathcal{T}$, $\delta$ its transition function and $m$ the least common multiple of the 0-cycles (cycles labelled with 0) in $\mathcal{A}$. For all $i$ large enough and congruent modulo $m$, the state $\delta(q, 10^n1)$ is independent from $i$, since we must finish every cycle. Thus $\mathcal{A}$ cannot distinguish large enough congruent blocks and we can build a (two-way) loop if it goes “too far”.

Lemma E.4. For all $k \geq 1$, $\pi^k \preceq_{1WFT} \pi$.

Proof. We rewrite $\pi$ as $\prod_{n=0}^{\infty} \prod_{j=0}^{k-1} 0^{kn+j}1$. A one-way transducer outputting $(0^n1)^k$ when reading $\prod_{j=0}^{k-1} 0^{kn+j}1$ can be built.