Untwisting two-way transducers in elementary time

Félix Baschenis, Olivier Gauwin, Anca Muscholl, Gabriele Puppis

Untwisting two-way transducers in elementary time

Félix Baschenis
Université de Bordeaux, LaBRI
felix.baschenis@labri.fr

Olivier Gauwin
Université de Bordeaux, LaBRI
olivier.gauwin@labri.fr

Anca Muscholl
Université de Bordeaux, LaBRI
anca@labri.fr

Gabriele Puppis
CNRS, LaBRI
gabriele.puppis@labri.fr

***Abstract*—Functional transductions realized by two-way transducers (equivalently, by streaming transducers and by MSO transductions) are the natural and standard notion of "regular" mappings from words to words. It was shown recently (LICS'13) that it is decidable if such a transduction can be implemented by some one-way transducer, but the given algorithm has non-elementary complexity. We provide an algorithm of different flavor solving the above question, that has double exponential space complexity. We further apply our technique to decide whether the transduction realized by a two-way transducer can be implemented by a sweeping transducer, with either known or unknown number of passes.**

I. Introduction

Since the early times of computer science, transducers have been identified as a fundamental notion of computation, where one is interested how objects can be transformed into each other. Numerous fields of computer science are ultimately concerned with transformations, ranging from databases to image processing, and an important issue is to perform transformations with low costs, whenever possible.

The most basic form of transformers are devices that process an input and produce outputs during the processing, using finite memory. Such devices are called finite-state transducers. Word-to-word finite-state transducers were considered in very early work in formal language theory [1, 2, 3], and it was soon clear that they are much more challenging than finite-state word acceptors - the classical finite-state automata. One essential difference between transducers and automata over words is that the capability to process the input in both directions strictly increases the expressive power in the case of transducers, whereas this does not for automata [4, 5]. In other words, two-way word transducers are strictly more expressive than one-way word transducers.

We consider in this paper functional transducers, that compute functions from words to words. Two-way word transducers capture very nicely the notion of regularity in this setting. Regular word functions, i.e. functions computed by functional two-way transducers, inherit many of the characterizations and algorithmic properties of the robust class of regular languages. Engelfriet and Hoogeboom [6] showed that monadic second-order definable graph transductions, restricted to words, are equivalent to two-way transducers — this justifies the notation "regular" word functions, in the spirit of classical results in automata theory and logic by Büchi, Elgot, Rabin and others. Recently, Alur and Cerný [7] proposed an enhanced version of one-way transducers called streaming transducers, and showed that they are equivalent to the two previous models. A streaming transducer processes the input word from left to right, and stores (partial) output words in finitely many, write-only registers.

Two-way transducers raise challenging questions about resource requirements. One crucial resource is the number of times the transducer needs to re-process the input word. In particular, the case where the input can be processed in a single pass, from left to right, is very attractive as it corresponds to the setting of *streaming*, where the (possibly very large) inputs do not need to be stored in order to be processed. Recently, it was shown in [8] that it is decidable whether the transduction defined by a functional two-way transducer can be implemented by a one-way transducer. However, the decision procedure of [8] has non-elementary complexity, and it is very natural to ask whether one can do better. We gave in [9, 10] an exponential space algorithm in the special case of *sweeping* transducers: head reversals are only allowed at the extremities of the input. However, sweeping transducers are known to be strictly less expressive than two-way transducers.

In this paper we provide an algorithm of elementary complexity for deciding whether the transduction defined by a functional two-way transducer can be implemented by a one-way transducer: the decision algorithm has double exponential space complexity, and an equivalent one-way transducer (if it exists), can be constructed with triple exponential size. The known lower bound [9] is double exponential size. Our techniques can be further adapted to characterize definability of transductions by other models of transducers, e.g. to characterize sweeping transducers within the class of two-way transducers.

Related work. Besides the papers mentioned above, there are several recent results around the expressivity and the resources of two-way transducers, or equivalently, streaming transducers. First-order definable transductions were shown to be equivalent to transductions defined by aperiodic streaming transducers [11] and to aperiodic two-way transducers [12]. An effective characterization of aperiodicity for one-way transducers was obtained in [13].

In [10, 14] the minimization of the number of registers of deterministic streaming transducers, resp., passes of functional sweeping transducers, was shown to be decidable. An algebraic characterization of (not necessarily functional) two-way transducers over unary alphabets was provided in [15]. It was shown that in this case sweeping transducers have the same expressivity. The expressivity of non-deterministic input-unary or output-unary two-way transducers was investigated in [16].

Overview. Section II introduces basic notations for two-way transducers, and Section III states the main result. Section IV is devoted to the effect of pumping runs on outputs, and Section V introduces the main tool for our characterization. Section VI handles the construction of an equivalent one-way transducer. Finally, Section VII describes a procedure to decide whether a functional transducer is equivalent to a sweeping transducer.

II. Preliminaries

Two-way automata and transducers. We start with some basic notations and definitions for two-way automata (resp., transducers). We assume that every input word $u = a_1 \cdots a_n$ has two special delimiting symbols $a_1 = \vdash$ and $a_n = \dashv$ that do not occur elsewhere: $a_i \notin \{\vdash, \dashv\}$ for all $i = 2, \ldots, n-1$.

A *two-way automaton* $\mathcal{A} = \langle Q, \Sigma, \vdash, \dashv, \delta, q_0, F\rangle$ has a finite state set Q, input alphabet Σ, transition relation $\delta \subseteq Q \times (\Sigma \cup \{\vdash, \dashv\}) \times Q \times \{\mathsf{left}, \mathsf{right}\}$, initial state $q_0 \in Q$, and set of final states $F \subseteq Q$. By convention, left transitions on $\vdash$ are not allowed. A *configuration* of $\mathcal{A}$ has the form $u\,q\,v$, with $uv \in \{\vdash\}\cdot\Sigma^*\cdot\{\dashv\}$ and $q \in Q$. A configuration $u\,q\,v$ represents the situation where the current state of $\mathcal{A}$ is q and its head reads the first symbol of v (on input uv). If $(q, a, q', \mathsf{right}) \in \delta$, then there is a transition from any configuration of the form $u\,q\,av$ to the configuration $ua\,q'\,v$, which we denote $u\,q\,av \xrightarrow{a,\mathsf{right}} ua\,q'\,v$. Similarly, if $(q, a, q', \mathsf{left}) \in \delta$, then there is a transition from any configuration of the form $ub\,q\,av$ to the configuration $u\,q'\,bav$, denoted as $ub\,q\,av \xrightarrow{a,\mathsf{left}} u\,q'\,bav$. A *run* on w is a sequence of transitions. It is *successful* if it starts in the initial configuration $q_0\,w$ and ends in a configuration $w\,q$ with $q \in F$ — note that this latter configuration does not allow additional transitions. The *language* of $\mathcal{A}$ is the set of input words that admit a successful run of $\mathcal{A}$.

The definition of *two-way transducers* is similar to that of two-way automata, with the only difference that now there is an additional output alphabet Γ and the transition relation is a finite subset of $Q \times (\Sigma \cup \{\vdash, \dashv\}) \times \Gamma^* \times Q \times \{\mathsf{left}, \mathsf{right}\}$, which associates an output over Γ with each transition of the underlying two-way automaton. Formally, given a two-way transducer $\mathcal{T} = \langle Q, \Sigma, \vdash, \dashv, \Gamma, \delta, q_0, F\rangle$, we have a transition of the form $ub\,q\,av \xrightarrow{a,d|w} u'\,q'\,v'$, outputting w, whenever $(q, a, w, q', d) \in \delta$ and either $u' = uba$, $v' = v$ or $u' = u$, $v' = bav$, depending on whether $d = \mathsf{right}$ or $d = \mathsf{left}$. The *output* associated with a run $\rho = u_1\,q_1\,v_1 \xrightarrow{a_1,d_1|w_1} \ldots \xrightarrow{a_n,d_n|w_n} u_{n+1}\,q_{n+1}\,v_{n+1}$ of $\mathcal{T}$ is the word $\mathsf{out}(\rho) = w_1 \cdots w_n$. A transducer $\mathcal{T}$ defines a relation consisting of all pairs (u, w) such that $w = \mathsf{out}(\rho)$, for some successful run ρ on u.

The *domain* of $\mathcal{T}$, denoted $\mathsf{dom}(\mathcal{T})$, is the set of input words that have a successful run. For transducers $\mathcal{T}, \mathcal{T}'$, we write $\mathcal{T}' \subseteq \mathcal{T}$ to mean that $\mathsf{dom}(\mathcal{T}') \subseteq \mathsf{dom}(\mathcal{T})$ and the transductions computed by $\mathcal{T}, \mathcal{T}'$ coincide on $\mathsf{dom}(\mathcal{T}')$.

We say that $\mathcal{T}$ is *functional* if for each input u, at most one output w can be produced by any possible successful run on u. Finally, we say that $\mathcal{T}$ is *one-way* if it does not have transition rules of the form $(q, a, w, q', \mathsf{left})$.

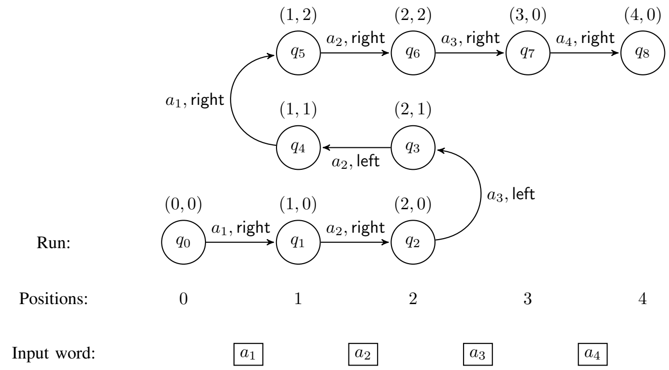


Fig. 1. Graphical presentation of a run by means of crossing sequences.

Crossing sequences. The first basic notion is that of crossing sequence. We follow the convenient presentation from [17], which appeals to a graphical representation of runs of a two-way transducer where each configuration is seen as point (location) in a two-dimensional space. Let $u = a_1 \cdots a_n$ be an input word (recall that $a_1 = \vdash$ and $a_n = \dashv$) and let ρ be a run of a two-way automaton (or transducer) $\mathcal{T}$ on u. The *positions* of ρ are the numbers from 0 to n, corresponding to "cuts" between two consecutive letters of the input. For example, position 0 is just before the first letter a_1, position n is just after the last letter a_n, and any other position x, with $1 \leqslant x < n$, is between the letters a_x and a_{x+1}. We say that a transition $u\,q\,v \xrightarrow{a,d} u'\,q'\,v'$ of ρ *crosses position* x if either $d = \mathsf{right}$ and $|u| = x$, or $d = \mathsf{left}$ and $|u'| = x$. A *location* of ρ is any pair (x, y) for which there are at least $y+1$ transitions in ρ crossing position x; the component y of a location is called *level*. Each location is associated a state. Formally, we say that q is the *state at location* $\ell = (x, y)$ in ρ, and we denote this by writing $\rho(\ell) = q$, if the $(y+1)$-th transition that crosses x ends up in state q. The *crossing sequence* at position x of ρ is the tuple $\rho|x = (q_0, \ldots, q_h)$, where the q_y's are all the states at locations of the form (x, y), for $y = 0, \ldots, h$.

As suggested by Fig. 1, any run can be represented as an annotated path between locations. For example, if a location (x, y) is reached by a rightward transition, then the head of the automaton has read the symbol a_x; if it is reached by a leftward transition, then the head has read the symbol a_{x+1}. Note that in a successful run ρ every crossing sequence has odd length and every rightward (resp. leftward) transition reaches a location with even (resp. odd) level. We can identify four types of transitions between locations, depending on the parities of the levels (the reader may refer again to Fig. 1):

$$(x, 2y) \xrightarrow{a_{x+1},\,\mathsf{right}} (x+1, 2y') \qquad \begin{array}{l} (x, 2y+1) \\ (x, 2y) \end{array} \Big) \; a_{x+1}, \mathsf{left}$$

$$(x, 2y+1) \xleftarrow{a_{x+1},\,\mathsf{left}} (x+1, 2y'+1) \qquad a_x, \mathsf{right} \; \Big(\begin{array}{l} (x, 2y+2) \\ (x, 2y+1) \end{array}$$

Hereafter, we will identify runs with the corresponding annotated paths between locations. It is also convenient to define a total order $\trianglelefteq$ on the locations of a run ρ by letting $\ell_1 \trianglelefteq \ell_2$ if ℓ_2 is reachable from ℓ_1 by following the path described by ρ — the order $\trianglelefteq$ on locations is called *run order*. Given two locations $\ell_1 \trianglelefteq \ell_2$ of a run ρ, we write $\rho[\ell_1, \ell_2]$ for the factor of the run that starts in ℓ_1 and ends in ℓ_2. Note that the latter is also a run and hence the notation $\mathrm{out}\big(\rho[\ell_1, \ell_2]\big)$ is permitted. Two runs ρ_1, ρ_2 can be concatenated, provided that ρ_1 ends in location (x, y), ρ_2 starts in location (x, y'), such that $y' = y \pmod 2$ and (x, y), (x, y') are labelled by the same state. We denote by $\rho_1\rho_2$ the run resulting from concatenating ρ_1 with ρ_2. Clearly, we have $\rho[\ell_1, \ell_2]\ \rho[\ell_2, \ell_3] = \rho[\ell_1, \ell_3]$ for all locations $\ell_1 \trianglelefteq \ell_2 \trianglelefteq \ell_3$.

Normalization. Without loss of generality, we will assume that successful runs of functional transducers are *normalized*, meaning that they never visit two locations with the same position, the same state, and both either at even or at odd level. Indeed, if this were not the case, say if a successful run ρ visited two locations $\ell_1 = (x, y)$ and $\ell_2 = (x, y')$ such that $\rho(\ell_1) = \rho(\ell_2)$ and y, y' are both even or both odd, then the output produced by ρ between ℓ_1 and ℓ_2 should be empty, as otherwise by repeating the factor $\rho[\ell_1, \ell_2]$ of ρ we could obtain successful runs that produces different outputs on the same input, thus contradicting the assumption that the transducer is functional. Now that we know that the output of ρ produced between ℓ_1 and ℓ_2 is empty, we could drop the factor $\rho[\ell_1, \ell_2]$, thus obtaining a successful run with the same output. It is easy to see that, in every normalized successful run, the crossing sequences have length at most $2|Q| - 1$.

We define $h_{\mathsf{max}} = 2|Q| - 1$. Moreover, by c_{max} we denote the *capacity* of the transducer, which is the maximal length of the output of a transition.

III. Two-way transducers vs one-way transducers

In this section we state our main result, which is the existence of an elementary algorithm for checking whether a two-way transducer is equivalent to some one-way transducer. We call such transducers *one-way definable*. Before stating our result, we give a few examples.

Example 1. *We consider two-way transducers that accept any input u from a given regular language R and output the word $u\,u$. We will argue how, depending on R, these transducers may or may not be one-way definable.*

1) *If $R = (a+b)^*$ there is no equivalent one-way transducer, as the output language is not regular. If R is finite, then the transduction mapping $u \in R$ to $u\,u$ can be implemented by a one-way transducer that guesses u (this requires as many states as the size of R), checks the input, and outputs two copies of the guessed word.*
2) *A special case of transduction with finite domain is given by $R_n = \{a_0\, w_0 \cdots a_{2^n-1}\, w_{2^n-1} \ :\ a_0, \dots, a_{2^n-1} \in \{a, b\}\}$, where $n \in \mathbb{N}$ and each w_i is the binary encoding of the counter $i = 0, \dots, 2^n - 1$. It is easy to see (cf. Proposition 15 [9]) that the transduction mapping $u \in R_n$ to $u\,u$ can be implemented by a two-way transducer with quadratically many states w.r.t. n, while every equivalent one-way transducer has at least 2^{2^n} states, since it needs to guess a word of length 2^n.*
3) *Consider now the periodic language $R = (abc)^*$. The function that maps $u \in R$ to $u\,u$ can be easily implemented by a one-way transducer: it suffices to output two letters (i.e., ab, ca, bc, in turn) for each input letter, while checking that the input is in R.*

Example 2. *We consider a slightly more complicated transduction that is defined on input words of the form $u_1 \# \dots \# u_n$, where each factor u_i is over the alphabet $\Sigma = \{a, b, c\}$. The output of the transduction is of the form $w_1 \# \dots \# w_n$, where each w_i is either $u_i\, u_i$ or just u_i, depending on whether or not $u_i \in (abc)^*$ and u_{i+1} has even length, with $u_{n+1} = \varepsilon$.*

The obvious way to implement the transduction is by means of a two-way transducer that performs multiple passes on the factors of the input: a first left-to-right pass is performed on $u_i \# u_{i+1}$ to produce the first copy of u_i and to check whether $u_i \in (abc)^*$ and $|u_{i+1}|$ is even; if so, a second pass on u_i is performed to produce another copy of u_i.

The transduction can also be implemented by a one-way transducer: when entering a factor u_i, the transducer guesses whether or not $u_i \in (abc)^*$ and $|u_{i+1}|$ is even; depending on this it outputs either $(abc\,abc)^{\frac{|u_i|}{3}}$ or u_i, and checks that the guess is correct.

Our main result is:

Theorem 3. *There is an algorithm that from a functional two-way transducer $\mathcal{T}$ constructs in triple exponential time a* one-way *transducer $\mathcal{T}'$ with the following properties:*

- *$\mathcal{T}' \subseteq \mathcal{T}$,*
- *$\mathsf{dom}(\mathcal{T}) = \mathsf{dom}(\mathcal{T}')$ iff $\mathcal{T}$ is one-way definable.*

Moreover, the second property above can be checked in double exponential space w.r.t. $|\mathcal{T}|$.

We remark that a similar characterization for a much more restricted class of transducers (sweeping transducers) appeared in [9]. The proof of Theorem 3, however, is more technical, as it requires a better understanding of the structure of the runs of two-way transducers and a non-trivial generalization of the combinatorial arguments from [9].

The proof of the theorem spans along the next three sections. In Section IV, we present the basic concepts for reasoning on runs of two-way automata. This includes the definition of a finite semigroup for describing the shapes of two-way runs, as well as Ramsey-type arguments that are used to bound the length of the outputs produced by pieces of runs without loops. In Section V we provide the main combinatorial arguments for characterizing one-way definability. The crucial notion will be that of *inversion*, that captures behaviours of the two-way transducer that are problematic for one-way definability. Finally, in Section VI we exploit the combinatorial results and the Ramsey-type arguments to derive the existence of suitable decompositions of runs that lead to the construction of equivalent one-way transducers.

IV. Untangling runs of two-way transducers

This section is devoted to untangling the structure of runs of two-way transducers. Whereas the classical transformation of two-way automata into one-way automata based on crossing sequences is rather simple, we will need a much deeper understanding of runs of two-way transducers, because of the additional outputs. In a nutshell, being one-way definable is related to periodicities (with bounded periods) in the output, and these periodicities are generated by loops in the run. We will actually work with so called idempotent loops, that generate periodicities in the output in a "nice" way. We will derive the existence of idempotent loops with bounded outputs using Ramsey-based arguments.

We fix throughout the paper a functional two-way transducer $\mathcal{T}$, an input word u, and a successful run ρ of $\mathcal{T}$ on u. We assume that ρ is normalized, i.e., every state occurs at most once in each crossing sequence of ρ at levels of a given parity.

For simplicity, we denote by ω the length of the input word u. We will consider *intervals of positions* of the form $I = [x_1, x_2]$, with $0 \leqslant x_1 < x_2 \leqslant \omega$. The *containment relation* $\subseteq$ on intervals is defined by $[x_3, x_4] \subseteq [x_1, x_2]$ if $x_1 \leqslant x_3 < x_4 \leqslant x_2$.

Factors, flows, and effects. A factor of a run ρ is a contiguous subsequence of ρ. A factor *intercepted* by an interval $I = [x_1, x_2]$ is a maximal factor of ρ that visits only positions $x \in I$, and never uses a left transition from position x_1 or a right transition from position x_2.

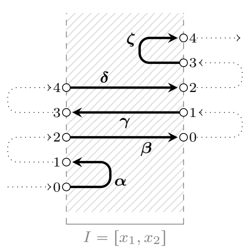


Fig. 2. Intercepted factors.

Fig. 2 on the right gives an example of an interval I that intercepts the factors $\alpha, \beta, \gamma, \delta, \zeta$. The numbers that annotate the endpoints of the factors represent their levels.

Every factor α intercepted by an interval $I = [x_1, x_2]$ is of one of the four types below, depending on its first location (x, y) and its last location (x', y'):

- α is an LL-factor if $x = x' = x_1$,
- α is an RR-factor if $x = x' = x_2$,
- α is an LR-factor if $x = x_1$ and $x' = x_2$,
- α is an RL-factor if $x = x_2$ and $x' = x_1$.

In Fig. 2 we see that α is an LL-factor, β, δ are LR-factors, ζ is an RR-factor, and γ is an RL-factor.

Definition 4. *Let ρ be a run and $I = [x_1, x_2]$ an interval of ρ. Let h_i be the length of the crossing sequence $\rho|x_i$ for both $i = 1$ and $i = 2$.*

The flow *F_I of I is a directed graph with set of nodes $\{0, \ldots, \max(h_1, h_2) - 1\}$ and set of edges consisting of all (y, y') such that there exists a factor of ρ intercepted by I that starts at location (x_i, y) and ends at location (x_j, y'), for $i, j \in \{1, 2\}$.*

The effect *E_I of I is the triple (F_I, c_1, c_2), where $c_i = \rho|x_i$ is the crossing sequence at x_i.*

For example, the interval I of Fig. 2 has the flow graph $0 \mapsto 1 \mapsto 3 \mapsto 4 \mapsto 2 \mapsto 0$. It is easy to see that every node of a flow F_I has at most one incoming and at most one outgoing edge. More precisely, if $y < h_1$ is even, then it has one outgoing edge (corresponding to an LR- or LL-factor intercepted by I), and if it is odd it has one incoming edge (corresponding to an RL- or LL-factor intercepted by I). Similarly, if $y < h_2$ is even, then it has one incoming edge (corresponding to an LR- or RR-factor), and if it is odd it has one outgoing edge (corresponding to an RL- or RR-factor).

In the following we consider generic effects that are not necessarily associated with intervals of specific runs. The definition of such effects should be clear: these are triples consisting of a graph (called flow) and two crossing sequences of lengths $h_1, h_2 \leqslant h_{\mathsf{max}}$, with sets of nodes of the form $\{0, \ldots, \max(h_1, h_2) - 1\}$, that satisfy the in/out-degree properties stated above.

It is convenient to distinguish the edges in a flow based on the parity of the source and target nodes. Formally, we partition any flow F into the following subgraphs:

- F_{LR} consists of all edges of F between pairs of even nodes,
- F_{RL} consists of all edges of F between pairs of odd nodes,
- F_{LL} consists of all edges of F from an even node to an odd node,
- F_{RR} consists of all edges of F from an odd node to an even node.

We denote by $\mathcal{F}$ (resp. $\mathcal{E}$) the set of all flows (resp. effects) augmented with a dummy element $\perp$. We equip both sets $\mathcal{F}$ and $\mathcal{E}$ with a semigroup structure, where the corresponding products $\circ$ and $\odot$ are defined below (similar definitions appear in [18]). We need this semigroup structure in order to identify *idempotent loops*, that play a crucial role in our characterization of one-way definability.

Definition 5. *For two graphs G, G', we denote by $G \cdot G'$ the graph with edges of the form (y, y'') such that (y, y') is an edge of G and (y', y'') is an edge of G', for some node y' that belongs to both G and G'. Similarly, we denote by G^* the graph with edges (y, y') such that there exists a (possibly empty) path in G from y to y'.*

The product of two flows F, F' is the unique flow $F \circ F'$ (if it exists) such that:

- $(F \circ F')_{\mathsf{LR}} = F_{\mathsf{LR}} \cdot (F'_{\mathsf{LL}} \cdot F_{\mathsf{RR}})^* \cdot F'_{\mathsf{LR}}$,
- $(F \circ F')_{\mathsf{RL}} = F'_{\mathsf{RL}} \cdot (F_{\mathsf{RR}} \cdot F'_{\mathsf{LL}})^* \cdot F_{\mathsf{RL}}$,
- $(F \circ F')_{\mathsf{LL}} = F_{\mathsf{LL}} \ \cup \ F_{\mathsf{LR}} \cdot (F'_{\mathsf{LL}} \cdot F_{\mathsf{RR}})^* \cdot F'_{\mathsf{LL}} \cdot F_{\mathsf{RL}}$,
- $(F \circ F')_{\mathsf{RR}} = F'_{\mathsf{RR}} \ \cup \ F'_{\mathsf{RL}} \cdot (F_{\mathsf{RR}} \cdot F'_{\mathsf{LL}})^* \cdot F_{\mathsf{RR}} \cdot F'_{\mathsf{LR}}$.

If no flow $F \circ F'$ exists with the above properties, then we let $F \circ F' = \perp$.

The product of two effects $E = (F, c_1, c_2)$ and $E' = (F', c'_1, c'_2)$ is either the effect $E \odot E' = (F \circ F', c_1, c'_2)$ or the dummy element $\perp$, depending on whether $F \circ F' \neq \perp$ and $c_2 = c'_1$.

For example, let F be the flow of interval I in Fig. 2. Then $(F \circ F)_{\mathsf{LL}} = \{(0, 1), (2, 3)\}$, $(F \circ F)_{\mathsf{RR}} = \{(1, 2), (3, 4)\}$, and

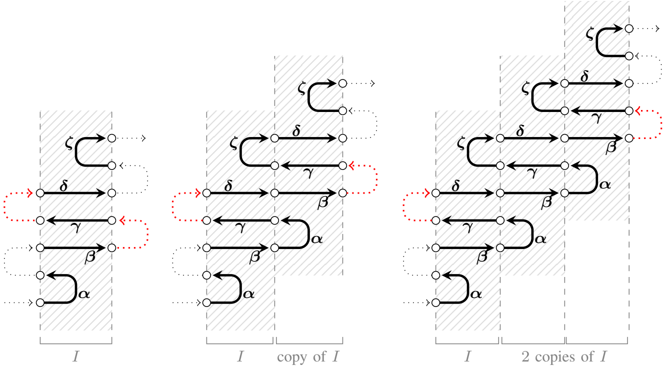


Fig. 3. Pumping a loop in a run.

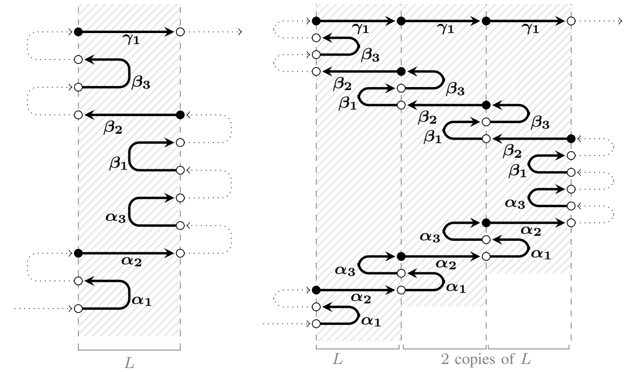


Fig. 4. Pumping an idempotent loop with three components.

$(F \circ F)_{\mathsf{LR}} = \{(4,0)\}$ — one can quickly verify this with the help of Fig. 3.

It is also easy to see that $(\mathcal{F}, \circ)$ and $(\mathcal{E}, \odot)$ are finite semigroups, and that for every run ρ and every pair of consecutive intervals $I = [x_1, x_2]$ and $J = [x_2, x_3]$ of ρ, $F_{I \cup J} = F_I \circ F_J$ and $E_{I \cup J} = E_I \odot E_J$. In particular, the function E that associates each interval I of ρ with the corresponding effect E_I can be seen as a semigroup homomorphism.

Note that, in a normalized successful run, there are at most $|Q|^{h_{\mathsf{max}}}$ distinct crossing sequences and at most $4^{h_{\mathsf{max}}}$ distinct flows, since there are at most h_{max} edges in a flow, and each one has one of the 4 possible types $\mathsf{LL}, \dots, \mathsf{RR}$. Hence there are at most $(2|Q|)^{2h_{\mathsf{max}}}$ distinct effects.

Loops and components. Loops of a two-way run are the basic building blocks for characterizing one-way definability. We will consider special types of loops, called idempotent loops, when showing that outputs generated in non left-to-right manner are essentially periodic.

Definition 6. *A* loop *of ρ is an interval $L = [x_1, x_2]$ whose endpoints have the same crossing sequences, i.e. $\rho|x_1 = \rho|x_2$. It is said to be* idempotent *if $E_L = E_L \odot E_L$ and $E_L \neq \perp$.*

For example, the interval I of Fig. 2 is a loop, if one assumes that the crossing sequences at the borders of I are the same. However, by comparing with Fig. 3, it is easy to see that I is not idempotent. On the other hand, the loop consisting of 2 copies of I is idempotent.

Given a loop $L = [x_1, x_2]$ and a number $m \in \mathbb{N}$, we can introduce m new copies of L and connect the intercepted factors in the obvious way. Fig. 3 shows how to do this for $m = 1$ and $m = 2$. The operation that we just described is called *pumping*, and results in a new run of the transducer $\mathcal{T}$ on the word

$$\mathsf{pump}_L^{m+1}(u) \ := \ u[0,x_1] \cdot \big(u[x_1+1, x_2]\big)^{m+1} \cdot u[x_2+1, n] \ .$$

We denote by $\mathsf{pump}_L^{m+1}(\rho)$ the pumped[1] run on $\mathsf{pump}_L^{m+1}(u)$.

The goal in this section is to describe the shape of the pumped run $\mathsf{pump}_L^{m+1}(\rho)$ (and the produced output as well) when L is an *idempotent* loop. We will focus on idempotent loops because pumping non-idempotent loops may induce permutations of factors that are difficult to handle. For example, if we consider again the non-idempotent loop I to the left of Fig. 3, the factor of the run between β and γ (to the right of I, highlighted in red) precedes the factor between γ and δ (to the left of I, again in red), but this ordering is reversed when a new copy of I is added.

When pumping a loop L, subsets of factors intercepted by L are glued together to form longer factors intercepted by the unioned copies of L. The concept of component that we introduce below aims at identifying the groups of factors that are glued together.

Definition 7. *A* component *of a loop L is any strongly connected component of its flow F_L (note that this is also a cycle, since every node in it has in/out-degree 1). Given a component C, we denote by $\min(C)$ (resp. $\max(C)$) the minimum (resp. maximum) node in C. We say that C is* left-to-right *(resp.* right-to-left*) if $\min(C)$ is even (resp., odd). An (L,C)-factor is a factor of the run that is intercepted by L and corresponds to an edge of C.*

For example, the loop I of Fig. 3 contains a single component $C = \{0 \mapsto 1 \mapsto 3 \mapsto 4 \mapsto 2 \mapsto 0\}$ which is left-to-right. Another example is given in Fig. 4, where the loop L has three components C_1, C_2, C_3 (ordered from bottom to top): $\alpha_1, \alpha_2, \alpha_3$ are the (L, C_1)-factors, $\beta_1, \beta_2, \beta_3$ are the (L, C_2)-factors, and γ_1 is the unique (L, C_3)-factor.

We will usually list the (L,C)-factors based on their order of occurrence in the run.

The following lemma (proved in the appendix) describes the precise shape and order of such factors when the loop L is idempotent. It can be used to reason on the shape of runs obtained by pumping idempotent loops.

Lemma 8. *If C is a left-to-right (resp. right-to-left) component of an idempotent loop L, then the (L,C)-factors are in the following order: k LL-factors (resp. RR-factors), followed by one LR-factor (resp. RL-factor), followed by k RR-factors (resp. LL-factors), for some $k \geqslant 0$.*

We also need to introduce the notions of *anchor* (Def. 9)

[1] Using similar constructions, one could remove a loop L from a run ρ, resulting in the run $\mathsf{pump}_L^0(\rho)$. As we do not need this, the operation pump_L will always be parametrized by a positive number $m+1$.

and *trace* (Def. 10).

Definition 9. *Let C be a component of an idempotent loop $L = [x_1, x_2]$. The* anchor *of C inside L, denoted*[2] $\mathsf{an}(C)$*, is either the location $\big(x_1, \max(C)\big)$ or the location $\big(x_2, \max(C)\big)$, depending on whether C is left-to-right or right-to-left.*

Intuitively, the anchor $\mathsf{an}(C)$ of a component C of L is the source location of the unique LR- or RL-factor intercepted by L that corresponds to an edge of C (recall Lemma 8).

Definition 10. *Let C be a component of some idempotent loop L and let $(i_0, i_1), (i_1, i_2), \ldots, (i_{k-1}, i_k), (i_k, i_{k+1})$ be a cycle of C, where $i_0 = i_{k+1} = \max(C)$. For every $j = 0, \ldots, k$, let β_j be the factor intercepted by L that corresponds to the edge (i_j, i_{j+1}) of C. The* trace *of C inside L is the run $\mathsf{tr}(C) = \beta_0\ \beta_1\ \cdots\ \beta_k$ (note that this is not necessarily a factor of the original run ρ).*

Intuitively, the trace $\mathsf{tr}(C)$ is obtained by concatenating the (L, C)-factors together, where the first factor is the (unique) LR-/RL-factor that starts at the anchor $\mathsf{an}(C)$ and the remaining ones are the LL-factors interleaved with the RR-factors.

For example, by referring again to the components C_1, C_2, C_3 of Fig. 4, we have the following traces: $\mathsf{tr}(C_1) = \alpha_2\ \alpha_1\ \alpha_3$, $\mathsf{tr}(C_2) = \beta_2\ \beta_1\ \beta_3$, and $\mathsf{tr}(C_3) = \gamma_1$.

As shown by the following proposition (proved in the appendix), iterations of idempotent loops translate to iterations of traces $\mathsf{tr}(C)$ of components.

Proposition 11. *Let L be an idempotent loop of ρ with components $C_1, \ldots, C_k$, listed according to the order of their anchors: $\mathsf{an}(C_1) \lhd \cdots \lhd \mathsf{an}(C_k)$. For all $m \in \mathbb{N}$, we have*

$$\mathsf{pump}_L^{m+1}(\rho) \;=\; \rho_0\ \mathsf{tr}(C_1)^m\ \rho_1\ \cdots\ \rho_{k-1}\ \mathsf{tr}(C_k)^m\ \rho_k$$

where

- *ρ_0 is the prefix of ρ that ends at $\mathsf{an}(C_1)$,*
- *ρ_i is the factor $\rho[\mathsf{an}(C_i), \mathsf{an}(C_{i+1})]$, for all $1 \leqslant i < k$,*
- *ρ_k is the suffix of ρ that starts at $\mathsf{an}(C_k)$.*

For example, referring to the left hand-side of Fig. 4, the run ρ_0 goes until the first location marked by a black dot. The run ρ_1 and ρ_2, resp., are between the first and the second black dot, and the second and third black dot. Finally, ρ_3 is the suffix starting at the last black dot. The pumped run $\mathsf{pump}_L^{m+1}(\rho)$ for $m = 2$ is depicted to the right of Fig. 4.

Ramsey-type arguments. We conclude the section by describing a technique that can be used for bounding the length of the outputs produced by factors of the run ρ. This technique is based on Ramsey-type arguments and relies on Simon's "factorization forest" theorem [19, 20], which we recall below.

Let X be a set of positions of ρ. A *factorization forest* for X is an unranked tree, where the nodes are intervals I with endpoints in X, labelled with the corresponding effect E_I, the ancestor relation is given by the containment order on intervals, the leaves are the minimal intervals $[x_1, x_2]$, with x_2 successor of x_1 in X, and for every internal node I with children $J_1, \ldots, J_k$, we have:

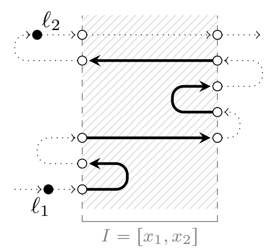
- $I = J_1 \cup \cdots \cup J_k$,
- $E_I = E_{J_1} \odot \cdots \odot E_{J_k}$,
- if $k > 2$, then $E_I = E_{J_1} = \cdots = E_{J_k}$ is an idempotent of the semigroup $(\mathcal{E}, \odot)$.

We will make use of the following three constants defined from the transducer $\mathcal{T}$: the maximum number c_{max} of letters output by a single transition, the maximal length $h_{\mathsf{max}} = 2|Q| - 1$ of a crossing sequence, and the maximal size $e_{\mathsf{max}} = (2|Q|)^{2h_{\mathsf{max}}}$ of the effect semigroup $(\mathcal{E}, \odot)$. By $\boldsymbol{B} = \boldsymbol{c_{\mathsf{max}}} \cdot \boldsymbol{h_{\mathsf{max}}} \cdot (\boldsymbol{2^{3e_{\mathsf{max}}}} + \boldsymbol{4})$ we will denote the main constant appearing in all subsequent sections.

Theorem 12 (Factorization forest theorem [19, 20])**.** *For every set X of positions of ρ, there is a factorization forest for X of height at most $3e_{\mathsf{max}}$.*

It is easy to use the above theorem to show that every run that produces an output longer than $\boldsymbol{B}$ contains an idempotent loop with non-empty output. Below, we present a result in the same spirit, but refined in a way that it can be used to find anchors of components of loops inside specific intervals.

In order to state it formally, we need to consider subsequences of ρ induced by sets of locations that are not necessarily intervals. Recall that $\rho[\ell_1, \ell_2]$ denotes the factor of ρ delimited by two locations $\ell_1 \trianglelefteq \ell_2$. Similarly, given any set Z of (possibly non-consecutive) locations, we denote by $\rho \mid Z$ the subsequence of ρ induced by Z. A transition of $\rho \mid Z$ is a transition from some ℓ to ℓ', where both ℓ, ℓ' belong to Z. The output $\mathsf{out}(\rho \mid Z)$ is the concatenation of the outputs of the transitions of $\rho \mid Z$ (in the order given by ρ). An example of subrun $\rho \mid Z$ is represented by the thick arrows in the figure to the right, where $Z = [\ell_1, \ell_2] \cap (I \times \mathbb{N})$.



Theorem 13. *Let $I = [x_1, x_2]$ be an interval of positions, $K = [\ell_1, \ell_2]$ an interval of locations, and $Z = K \cap (I \times \mathbb{N})$. If $\big|\mathsf{out}(\rho \mid Z)\big| > \boldsymbol{B}$, then there exist an idempotent loop L and a component C of L such that*

- $x_1 < \min(L) < \max(L) < x_2$ *(in particular, $L \subsetneq I$),*
- $\ell_1 \lhd \mathsf{an}(C) \lhd \ell_2$ *(in particular, $\mathsf{an}(C) \in K$),*
- $\mathsf{out}(\mathsf{tr}(C)) \neq \varepsilon$.

V. Inversions and periods

As suggested by Examples 1 and 2, a typical phenomenon that may prevent a transducer from being one-way definable is that of an *inversion*. An inversion essentially corresponds to a long output produced from right to left. The main result in this section is Proposition 16, that shows that the output produced between the locations delimiting an inversion must be periodic, with bounded period.

[2]In denoting the anchor — and similarly the trace — of a component C inside a loop L, we omit the annotation specifying L, since this is often understood from the context.

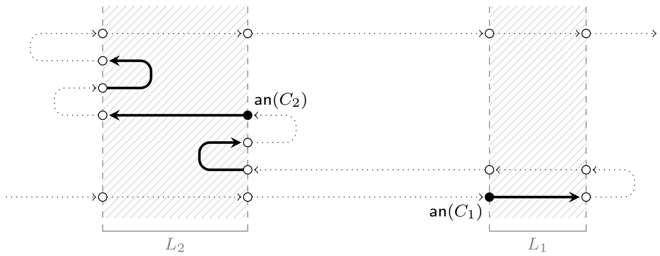


Fig. 5. An inversion with components intercepting the highlighted factors.

Definition 14. *An* inversion *of ρ is a tuple (L_1, C_1, L_2, C_2) such that*

- *L_i is an idempotent loop, for both $i = 1, 2$,*
- *C_i is a component of L_i, for both $i = 1, 2$,*
- $\mathsf{an}(C_1) \trianglelefteq \mathsf{an}(C_2)$,
- $\mathsf{an}(C_i) = (x_i, y_i)$*, for both $i = 1, 2$, and $x_1 \geqslant x_2$,*
- *both* $\mathsf{out}(\mathsf{tr}(C_1))$ *and* $\mathsf{out}(\mathsf{tr}(C_2))$ *are non-empty.*

Fig. 5 gives an example of an inversion involving the loop L_1 with its first component and the loop L_2 with its second component (we highlighted the anchors and the factors corresponding to these components).

Definition 15. *A word $w = a_1 \cdots a_n$ has* period *p if $a_i = a_{i+p}$ for all pairs of positions $i, i+p$ of w.*

For example, $w = abc\,abc\,ab$ has period 3.

One-way definability of functional two-way transducers essentially amounts to showing that the output produced by every inversion has bounded period. The proposition below shows a slightly stronger periodicity property, which refers to the output produced inside the inversion extended on both sides by the trace outputs. We will need this stronger property later, when dealing with overlapping portions of the run delimited by different inversions.

Proposition 16. *If $\mathcal{T}$ is one-way definable, then for every inversion (L_1, C_1, L_2, C_2) of a successful run ρ of $\mathcal{T}$, the word*

$$\mathsf{out}\big(\mathsf{tr}(C_1)\big)\ \mathsf{out}\big(\rho[\mathsf{an}(C_1), \mathsf{an}(C_2)]\big)\ \mathsf{out}\big(\mathsf{tr}(C_2)\big)$$

has period p that divides both $|\mathsf{out}(\mathsf{tr}(C_1))|$ and $|\mathsf{out}(\mathsf{tr}(C_2))|$. Moreover, $p \leqslant \boldsymbol{B}$.

The basic combinatorial argument for proving Proposition 16 is a classical result in word combinatorics called Fine and Wilf's theorem [21]. Essentially, the theorem says that, whenever two periodic words w_1, w_2 share a sufficiently long factor, then they have as period the greatest common divisor of the two original periods. Below, we state a slightly stronger variant of Fine-Wilf's theorem, which contains an additional claim showing how to align a common factor of the words w_1, w_2 so as to form a third word w_3 that contains a prefix of w_1 and a suffix of w_2. The additional claim will be fully exploited in the proof of Proposition 26.

Lemma 17 (Fine-Wilf's theorem)**.** *If $w_1 = w'_1\, w\, w''_1$ has period p_1, $w_2 = w'_2\, w\, w''_2$ has period p_2, and the common factor w has length at least $p_1 + p_2 - \gcd(p_1, p_2)$, then w_1, w_2, and $w_3 = w'_1\, w\, w''_2$ have period $\gcd(p_1, p_2)$.*

Two further combinatorial results are heavily used in the proof of Proposition 16. The first one is a result of Kortelainen [22], which was later improved and simplified by Saarela [23]. It is related to word equations with iterated factors, like those that arise from considering outputs of pumped versions of a run. To improve readability, we highlight the important iterations of factors inside the considered equations.

Theorem 18 (Theorem 4.3 in [23])**.** *Consider a word equation*

$$v_0\, \boldsymbol{v_1^m}\, v_2 \ldots v_{k-1}\, \boldsymbol{v_k^m}\, v_{k+1} \;=\; w_0\, \boldsymbol{w_1^m}\, w_2 \ldots w_{k'-1}\, \boldsymbol{w_{k'}^m}\, w_{k'+1}$$

where m is the unknown and v_i, w_j are words. Then the set of solutions of the equation is either finite or $\mathbb{N}$.

The second combinatorial result considers a word equation with iterated factors parametrized by two unknowns m_1, m_2 that occur in opposite order in the left, respectively right hand-side of the equation. This type of equation arises when we compare the output associated with an inversion of $\mathcal{T}$ and the output produced by an equivalent one-way transducer $\mathcal{T}'$.

Lemma 19. *Consider a word equation of the form*

$$v_0^{(m_1,m_2)}\, \boldsymbol{v_1^{m_1}}\, v_2^{(m_1,m_2)}\, \boldsymbol{v_3^{m_2}}\, v_4^{(m_1,m_2)} \;=\; w_0\, \boldsymbol{w_1^{m_2}}\, w_2\, \boldsymbol{w_3^{m_1}}\, w_4$$

where m_1, m_2 are the unknowns, v_1, v_3 are non-empty words, and $v_0^{(m_1,m_2)}, v_2^{(m_1,m_2)}, v_4^{(m_1,m_2)}$ are words that may contain factors of the form v^{m_1} or v^{m_2}, for a generic word v. If the above equation holds for all $m_1, m_2 \in \mathbb{N}$, then the words $\boldsymbol{v_1}\ \boldsymbol{v_1^{m_1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2}}\ \boldsymbol{v_3}$ are periodic with period $\gcd(|v_1|, |v_3|)$, for all $m_1, m_2 \in \mathbb{N}$.

The last ingredient used in the proof of Proposition 16 is a bound on the period of the output produced by an inversion. For this, we introduce a suitable notion of minimality of loops and loop components:

Definition 20. *Consider pairs (L, C) consisting of an idempotent loop L and a component C of L.*

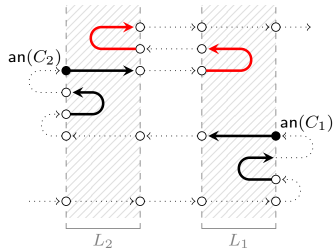
- *On such pairs, we define the relation $\sqsubset$ by $(L', C') \sqsubset (L, C)$ if $L' \subsetneq L$ and at least one (L', C')-factor is contained in some (L, C)-factor.*
- *A pair (L, C) is* output-minimal *if for all pairs $(L', C') \sqsubset (L, C)$, we have* $\mathsf{out}(\mathsf{tr}(C')) = \varepsilon$.

Note that the relation $\sqsubset$ is not a partial order in general (it is however antisymmetric). Lemma 21 below shows that the length of the output trace of C inside L is bounded whenever (L, C) is output-minimal.

Lemma 21. *For every output-minimal pair (L, C),* $|\mathsf{out}(\mathsf{tr}(C))| \leqslant \boldsymbol{B}$.

Proof sketch. We use a Ramsey-type argument here: if $|\mathsf{out}(\mathsf{tr}(C))| > \boldsymbol{B}$, then Theorem 13 can be applied to exhibit an idempotent loop strictly inside L and a component C of it with non-empty trace output. This would contradict the output-minimality of (L, C). $\square$

We remark that the above lemma cannot be used directly to bound the period of the output produced by an inversion. The reason is that we cannot assume that inversions are built up from output-minimal pairs. A counter-example is given in the figure to the right, which shows a run where the only inversion (L_1, C_1, L_2, C_2) contains pairs that are not output-minimal: the factors that produce long outputs are those in red, but they occur outside $\rho[\mathsf{an}(C_1), \mathsf{an}(C_2)]$.



We are now ready to prove Proposition 16. Here we only present the key ideas, and refer the reader to the appendix for more details.

Proof sketch of Proposition 16. In the first half of the proof we pump the two loops L_1 and L_2 so that we obtain also loops in the assumed equivalent one-way transducer $\mathcal{T}'$. We then consider the outputs of the pumped runs of $\mathcal{T}$ and $\mathcal{T}'$, which contain iterated factors parametrized by two natural numbers m_1, m_2. As those outputs must agree due to the equivalence of $\mathcal{T}, \mathcal{T}'$, we get an equation as in Lemma 19, where the word v_1 belongs to $\mathsf{out}(\mathsf{tr}(C_1))^+$ and the word v_3 belongs to $\mathsf{out}(\mathsf{tr}(C_2))^+$. Lemma 19 shows that the word described by the equation has period p dividing $\gcd(|v_1|, |v_3|)$, and Lemma 17 shows that p even divides $|\mathsf{out}(\mathsf{tr}(C_1))|$ and $|\mathsf{out}(\mathsf{tr}(C_2))|$. Finally, we use Theorem 18 to transfer the periodicity property from the word of the equation to the word $w = \mathsf{out}(\mathsf{tr}(C_1))\ \mathsf{out}(\rho[\mathsf{an}(C_1), \mathsf{an}(C_2)])\ \mathsf{out}(\mathsf{tr}(C_2))$ produced by the original run of $\mathcal{T}$. This is possible because the word of the equation is obtained by iterating factors of w. In particular, by reasoning separately on the parameters that define those iterations, and by stating the periodicity property as an equation in the form required by Theorem 18, one can prove that the periodicity equation holds on all parameters, and thus in particular on w.

In the second half of the proof we show that the period p is bounded by $\boldsymbol{B}$. This requires a refinement of the previous arguments and involves pumping the run of $\mathcal{T}$ simultaneously on three different loops. The idea is that by pumping we manage to find inversions with some output-minimal pair (L_0, C_0). In this way we show that the period p also divides $\mathsf{out}(\mathsf{tr}(C_0))$, which is bounded by $\boldsymbol{B}$ according to Lemma 21. □

VI. One-way definability

Proposition 16 is the main combinatorial argument for characterizing two-way transducers that are one-way definable. In this section we provide the remaining arguments. Roughly, the idea is to decompose every successful run ρ into factors that produce long outputs either in a left-to-right manner ("diagonals"), or based on an almost periodic pattern ("blocks").

We say that a word w is *almost periodic with bound* p if $w = w_0\ w_1\ w_2$ for some words w_0, w_2 of length at most p and some word w_1 of period at most p.

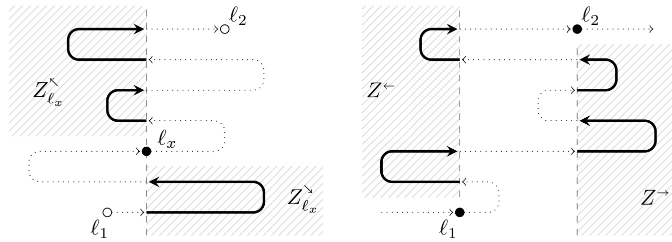


Fig. 6. Outputs that need to be bounded in a diagonal and in a block.

We illustrate the following definition in Fig. 6.

Definition 22. *Consider a factor $\rho[\ell_1, \ell_2]$ of the run, where $\ell_1 = (x_1, y_1)$, $\ell_2 = (x_2, y_2)$, and $x_1 \leqslant x_2$. We call $\rho[\ell_1, \ell_2]$*

- *a* diagonal *if for all $x \in [x_1, x_2]$, there is a location ℓ_x at position x such that $\ell_1 \trianglelefteq \ell_x \trianglelefteq \ell_2$ and the words $\mathsf{out}(\rho \mid Z^{\nwarrow}_{\ell_x})$ and $\mathsf{out}(\rho \mid Z^{\searrow}_{\ell_x})$ have length at most $\boldsymbol{B}$, where $Z^{\nwarrow}_{\ell_x} = [\ell_x, \ell_2] \cap \big([0, x] \times \mathbb{N}\big)$ and $Z^{\searrow}_{\ell_x} = [\ell_1, \ell_x] \cap \big([x, \omega] \times \mathbb{N}\big)$;*
- *a* block *if the word $\mathsf{out}(\rho[\ell_1, \ell_2])$ is almost periodic with bound $\boldsymbol{B}$, and $\mathsf{out}(\rho \mid Z^{\leftarrow})$ and $\mathsf{out}(\rho \mid Z^{\rightarrow})$ have length at most $\boldsymbol{B}$, where $Z^{\leftarrow} = [\ell_1, \ell_2] \cap \big([0, x_1] \times \mathbb{N}\big)$ and $Z^{\rightarrow} = [\ell_1, \ell_2] \cap \big([x_2, \omega] \times \mathbb{N}\big)$.*

Intuitively, the output of a diagonal $\rho[\ell_1, \ell_2]$ can be simulated while scanning the input interval $[x_1, x_2]$ from left to right, since the outputs of $\rho \mid Z^{\nwarrow}_{\ell_x}$ and $\rho \mid Z^{\searrow}_{\ell_x}$ are bounded. A similar argument applies to a block $\rho[\ell_1, \ell_2]$, where in addition, one exploits the fact that the output is almost periodic. Roughly, the idea is that one can simulate the output of a block by outputting symbols according to a periodic pattern, and in a number that is determined from the transitions on $u[x_1, x_2]$ and the guessed (bounded) outputs on $Z^{\leftarrow}$ and $Z^{\rightarrow}$.

The general idea for turning a two-way transducer $\mathcal{T}$ into an equivalent one-way transducer $\mathcal{T}'$ is to guess (and check) a factorization of a successful run of $\mathcal{T}$ into factors that are either diagonals or blocks, and properly arranged following the order of positions.

Definition 23. *A* decomposition *of ρ is a factorization $\prod_i \rho[\ell_i, \ell_{i+1}]$ of ρ into diagonals and blocks, where $\ell_i = (x_i, y_i)$ and $x_i < x_{i+1}$ for all i.*

The one-way transducer $\mathcal{T}'$ whose existence is stated by Theorem 3 simulates $\mathcal{T}$ precisely on those inputs u that have some successful run admitting a decomposition. To provide further intuition on the notion of decomposition, we consider again the transduction of Example 2 and the two-way transducer $\mathcal{T}$ that implements it in the most natural way. Fig. 7 shows an example of a run of $\mathcal{T}$ on an input of the form $u_1 \# u_2 \# u_3 \# u_4$, where $u_2, u_4 \in (abc)^*$, $u_1\, u_3 \notin (abc)^*$, and u_3 has even length. The factors of the run that produce long outputs are highlighted by the bold arrows. The first and third factors of the decomposition, i.e. $\rho[\ell_1, \ell_2]$ and $\rho[\ell_3, \ell_4]$, are diagonals (represented by the blue hatched areas); the second and fourth factors $\rho[\ell_2, \ell_3]$ and $\rho[\ell_4, \ell_5]$ are blocks (represented by the red hatched areas).

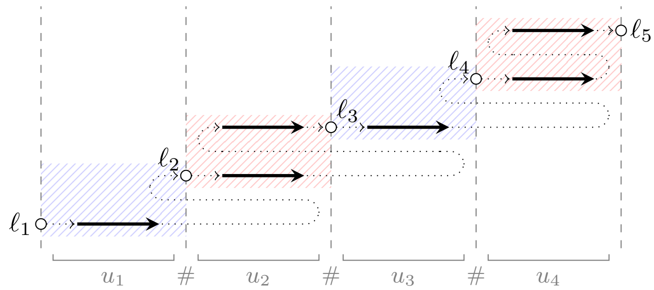


Fig. 7. A decomposition of a run of a two-way transducer.

Theorem 24. *Let $\mathcal{T}$ be a functional two-way transducer. The following are equivalent:*

P1) *$\mathcal{T}$ is one-way definable.*

P2) *For all inversions (L_1, C_1, L_2, C_2) of all successful runs of $\mathcal{T}$, the word*

$$\mathsf{out}\big(\mathsf{tr}(C_1)\big)\ \mathsf{out}\big(\rho[\mathsf{an}(C_1), \mathsf{an}(C_2)]\big)\ \mathsf{out}\big(\mathsf{tr}(C_2)\big)$$

has period $p \leqslant \boldsymbol{B}$ dividing $|\mathsf{out}(\mathsf{tr}(C_1))|$, $|\mathsf{out}(\mathsf{tr}(C_2))|$.

P3) *Every successful run of $\mathcal{T}$ admits a decomposition.*

The implication from **P1** to **P2** was already shown in Proposition 16. The rest of this section is devoted to prove the implications from **P2** to **P3** and from **P3** to **P1**. The issues related to the complexity of the characterization will be discussed further below.

From periodicity to existence of decompositions (P2→P3). As usual, we fix a successful run ρ of $\mathcal{T}$. We will prove a slightly stronger result than the implication from **P2** to **P3**, namely: if every inversion of ρ satisfies the periodicity property stated in **P2**, then ρ admits a decomposition (note that this is independent of whether other runs satisfy or not **P2**). To identify the blocks of a possible decomposition of ρ we consider a suitable equivalence relation between locations:

Definition 25. *A location ℓ is* covered *by an inversion (L_1, C_1, L_2, C_2) if $\mathsf{an}(C_1) \trianglelefteq \ell \trianglelefteq \mathsf{an}(C_2)$. We define the relation S by letting $\ell\ \mathsf{S}\ \ell'$ if ℓ, ℓ' are covered by the same inversion. We define the equivalence relation S^* as the reflexive and transitive closure of S.*

Locations covered by the same inversion (L_1, C_1, L_2, C_2) yield an interval w.r.t. the run ordering $\trianglelefteq$. Thus every non-singleton S^*-class can be seen as a union of such intervals, say $K_1, \dots, K_m$, that are two-by-two overlapping, namely, $K_i \cap K_{i+1} \neq \varnothing$ for all $i < m$. In particular, a non-singleton S^*-class is an interval of locations witnessed by a series of inversions $(L_{2i}, C_{2i}, L_{2i+1}, C_{2i+1})$ such that $\mathsf{an}(C_{2i}) \trianglelefteq \mathsf{an}(C_{2i+2}) \trianglelefteq \mathsf{an}(C_{2i+1}) \trianglelefteq \mathsf{an}(C_{2i+3})$.

The next result exploits the shape of a non-singleton S^*-class, the assumption that ρ satisfies the periodicity property stated in **P2**, and Lemma 17, to show that the output produced inside an S^*-class has bounded period.

Proposition 26. *If ρ satisfies the periodicity property stated in* **P2** *and $\ell \trianglelefteq \ell'$ are two locations in the same S^*-class, then $\mathsf{out}\big(\rho[\ell, \ell']\big)$ has period at most $\boldsymbol{B}$.*

The S^*-classes considered so far cannot be directly used as blocks for the desired decomposition of ρ, since the x-coordinates of their endpoints might not be in the appropriate order. The next definition takes care of this, by enlarging the S^*-classes according to x-coordinates of anchors.

Definition 27. *Let $K = [\ell, \ell']$ be a non-singleton S^*-class, let $\mathsf{an}(K)$ be the restriction of K to the locations that are anchors of components of inversions, and let $X_{\mathsf{an}(K)} = \{x : \exists y\ (x, y) \in \mathsf{an}(K)\}$ be the projection of $\mathsf{an}(K)$ on positions. We define $\mathsf{block}(K) = [\ell_1, \ell_2]$, where*

- *ℓ_1 is the latest location $(x, y) \trianglelefteq \ell$ such that $x = \min\big(X_{\mathsf{an}(K)}\big)$,*
- *ℓ_2 is the earliest location $(x, y) \trianglerighteq \ell'$ such that $x = \max\big(X_{\mathsf{an}(K)}\big)$*

(note that the location ℓ_1 exists since ℓ is the anchor of the first component of an inversion, and ℓ_2 exists for similar reasons).

Lemma 28. *If $K = [\ell, \ell']$ is a non-singleton S^*-class, then $\rho[\ell_1, \ell_2]$ is a block, where $[\ell_1, \ell_2] = \mathsf{block}(K)$.*

Proof sketch. The periodicity of $\mathsf{out}(\rho[\ell, \ell'])$ is obtained by applying Proposition 26. Then Theorem 13 is applied twice: first to bound $\mathsf{out}(\rho[\ell_1, \ell])$ and $\mathsf{out}(\rho[\ell', \ell_2])$ (hence proving that $\mathsf{out}(\rho[\ell_1, \ell_2])$ is almost periodic with bound $\boldsymbol{B}$), and second, to bound $\mathsf{out}(\rho \mid Z^{\leftarrow})$ and $\mathsf{out}(\rho \mid Z^{\rightarrow})$, as introduced in Definition 22. ☐

The next lemma shows that blocks do not overlap along the input axis:

Lemma 29. *Suppose that K_1 and K_2 are two different non-singleton S^*-classes such that $\ell \triangleleft \ell'$ for all $\ell \in K_1$ and $\ell' \in K_2$. Let $\mathsf{block}(K_1) = [\ell_1, \ell_2]$ and $\mathsf{block}(K_2) = [\ell_3, \ell_4]$, with $\ell_2 = (x_2, y_2)$ and $\ell_3 = (x_3, y_3)$. Then $x_2 < x_3$.*

Proof sketch. If $x_2 \geqslant x_3$, one can exhibit an inversion between a component of a loop in K_1 and another one in K_2, and deduce that $K_1 = K_2$. ☐

For the sake of brevity, we call S^**-block* any factor of the form $\rho \mid \mathsf{block}(K)$ that is obtained by applying Definition 27 to a non-singleton S^*-class K. The results obtained so far imply that every location covered by an inversion is also covered by an S^*-block (Lemma 28), and that the order of occurrence of S^*-blocks is the same as the order of positions (Lemma 29). So the S^*-blocks can be used as factors for the decomposition of ρ we are looking for. Below, we show that the remaining factors of ρ, which do not overlap the S^*-blocks, are diagonals. This will complete the construction of a decomposition of ρ.

Formally, we say that a factor $\rho[\ell_1, \ell_2]$ *overlaps* another factor $\rho[\ell_3, \ell_4]$ if $[\ell_1, \ell_2] \cap [\ell_3, \ell_4] \neq \varnothing$, $\ell_2 \neq \ell_3$, and $\ell_1 \neq \ell_4$.

Lemma 30. *Let $\rho[\ell_1, \ell_2]$ be a factor of ρ that does not overlap any S^*-block, with $\ell_1 = (x_1, y_1)$, $\ell_2 = (x_2, y_2)$, and $x_1 < x_2$. Then $\rho[\ell_1, \ell_2]$ is a diagonal.*

Proof sketch. If $\rho[\ell_1, \ell_2]$ is not a diagonal, we can find a location $\ell_1 \trianglelefteq \ell \trianglelefteq \ell_2$ for which $|\mathsf{out}(\rho \mid Z^{\nwarrow}_{\ell})| > \boldsymbol{B}$ and

$|\mathsf{out}(\rho \mid Z_{\ell}^{\searrow})| > \boldsymbol{B}$ (recall Definition 22). By applying again Theorem 13, we derive the existence of an inversion between ℓ_1 and ℓ_2, and thus of an S^*-block overlapping $\rho[\ell_1, \ell_2]$. $\square$

From decompositions to one-way definability (P3→P1). Hereafter, we denote by U the language of words $u \in \mathsf{dom}(\mathcal{T})$ such that *all* successful runs of $\mathcal{T}$ on u admit a decomposition.

So far, we know that if $\mathcal{T}$ is one-way definable (**P1**), then $U = \mathsf{dom}(\mathcal{T})$ (**P3**). This reduces the one-way definability problem for $\mathcal{T}$ to the containment problem $\mathsf{dom}(\mathcal{T}) \subseteq U$. We will see later how the latter problem can be decided in double exponential space by further reducing it to checking the emptiness of the intersection of the languages $\mathsf{dom}(\mathcal{T})$ and $U^{\complement}$, where $U^{\complement}$ is the complement of U.

Below, we show how to construct a one-way transducer $\mathcal{T}'$ of triple exponential size such that

$$\mathcal{T}' \subseteq \mathcal{T} \qquad \text{and} \qquad \mathsf{dom}(\mathcal{T}') \supseteq U.$$

In particular, the existence of such a transducer $\mathcal{T}'$ proves the implication from **P3** to **P1** of Theorem 24. It also proves the second item of Theorem 3, because when $\mathcal{T}$ is one-way definable, $U = \mathsf{dom}(\mathcal{T})$, and hence $\mathcal{T}$ and $\mathcal{T}'$ are equivalent.

Intuitively, given an input u, the one-way transducer $\mathcal{T}'$ will guess a successful run ρ of $\mathcal{T}$ on u and a decomposition of ρ, and then use the decomposition to simulate the output produced by ρ. Note that $\mathcal{T}'$ accepts at least all the words of U, possibly more. As a matter of fact, it would be difficult to construct a transducer whose domain coincides with U, since checking membership in U involves a universal quantification. The proof of the following result is in the appendix.

Proposition 31. *Given a functional two-way transducer $\mathcal{T}$, one can construct in* 3EXPTIME *a one-way transducer $\mathcal{T}'$ such that $\mathcal{T}' \subseteq \mathcal{T}$ and $\mathsf{dom}(\mathcal{T}') \supseteq U$.*

Deciding one-way definability. Recall that $\mathcal{T}$ is one-way definable iff $\mathsf{dom}(\mathcal{T}) \subseteq U$, so iff $\mathsf{dom}(\mathcal{T}) \cap U^{\complement} = \varnothing$. The lemma below exploits the characterization of Theorem 24 to show that the language $U^{\complement}$ can be recognized by an NFA $\mathcal{U}^{\complement}$ of triple exponential size. The lemma actually shows that the NFA recognizing $U^{\complement}$ can be constructed using double exponential *workspace*.

Lemma 32. *Given a functional two-way transducer $\mathcal{T}$, one can construct in* 2EXPSPACE *an NFA recognizing $U^{\complement}$.*

Proof. Consider an input word u. By Theorem 24 we know that $u \in U^{\complement}$ iff there exist a successful run ρ of $\mathcal{T}$ on u and an inversion $\mathcal{I} = (L_1, C_1, L_2, C_2)$ of ρ such that no positive number $p \leqslant \boldsymbol{B}$ is a period of the word

$$w_{\rho,\mathcal{I}} \;=\; \mathsf{out}\big(\mathsf{tr}(C_1)\big)\ \mathsf{out}\big(\rho[\mathsf{an}(C_1), \mathsf{an}(C_2)]\big)\ \mathsf{out}\big(\mathsf{tr}(C_2)\big).$$

The latter condition on $w_{\rho,\mathcal{I}}$ can be rephrased as follows: there is a function $f : \{1, \dots, \boldsymbol{B}\} \to \{1, \dots, |w_{\rho,\mathcal{I}}|\}$ such that $w_{\rho,\mathcal{I}}[f(p)] \neq w_{\rho,\mathcal{I}}[f(p) + p]$ for all positive numbers $p \leqslant \boldsymbol{B}$. Recall that $\boldsymbol{B} = c_{\mathsf{max}} \cdot h_{\mathsf{max}} \cdot (2^{3e_{\mathsf{max}}} + 4)$, where $h_{\mathsf{max}} = 2|Q| - 1$, $e_{\mathsf{max}} = (2|Q|)^{2h_{\mathsf{max}}}$, and Q is the state space of the two-way transducer $\mathcal{T}$. This means that the run ρ, the inversion $\mathcal{I}$, and the function f described above can all be guessed within double exponential space, namely, using a number of states that is at most a triple exponential w.r.t. $|\mathcal{T}|$. In particular, we can construct in 2EXPSPACE an NFA recognizing $U^{\complement}$. $\square$

As a consequence of the previous lemma and of Theorem 24, we have that the emptiness of the language $\mathsf{dom}(\mathcal{T}) \cap U^{\complement}$, and hence the one-way definability of $\mathcal{T}$, can be decided in 2EXPSPACE:

Corollary 33. *The problem of deciding whether a functional two-way transducer is one-way definable is in* 2EXPSPACE.

VII. Definability by sweeping transducers

A two-way transducer is called *sweeping* if every successful run of it performs reversals only at the extremities of the input word, i.e. when reading the symbols $\vdash$ or $\dashv$. Similarly, we call it *k-pass sweeping* if it is sweeping and every successful run performs at most $k - 1$ reversals. Clearly, a 1-pass sweeping transducer is the same as a one-way transducer.

In this section we are considering the following question: given a functional two-way transducer, is it equivalent to some k-pass sweeping transducer? We call such transducers *k-pass sweeping definable*. If the parameter k is not given *a priori*, then we denote them as *sweeping definable transducers*.

In [10] we built up on the characterization of one-way definability for (the restricted class of) sweeping transducers [9] in order to determine the minimal number of passes required by sweeping transductions. Essentially, the idea was to consider a generalization of the notion of inversion, called *k-inversion*, and proving that k-pass sweeping definability is equivalent to asking that every k-inversion generates a periodic output.

We show that we can follow the same approach for two-way transducers. More precisely, we first define a *co-inversion* in a way similar to Definition 14, namely, as a tuple (L_1, C_1, L_2, C_2) consisting of two idempotent loops L_1, L_2, a component C_1 of L_1, and a component C_2 of L_2 such that

- $\mathsf{an}(C_1) \trianglelefteq \mathsf{an}(C_2)$,
- $\mathsf{out}(\mathsf{tr}(C_1)), \mathsf{out}(\mathsf{tr}(C_2)) \neq \varepsilon$, and
- $\mathsf{an}(C_i) = (x_i, y_i)$ for $i = 1, 2$, then $x_1 \leqslant x_2$.

The only difference compared to inversions is the ordering of the positions of the anchors, which is now reversed.

Alternating inversions and co-inversions leads to:

Definition 34. *A* k-inversion *is a tuple $\overline{\mathcal{I}} = (\mathcal{I}_0, \dots, \mathcal{I}_{k-1})$, where $\mathcal{I}_i = (L_i, C_i, L'_i, C'_i)$ is either an inversion or a co-inversion depending on whether i is even or odd, and $\mathsf{an}(C'_i) \trianglelefteq \mathsf{an}(C_{i+1})$ for all $i < k - 1$.*
A k-inversion $\overline{\mathcal{I}}$ is safe *if for some $0 \leqslant i < k$, the word*

$$\mathsf{out}\big(\mathsf{tr}(C_i)\big)\ \mathsf{out}\big(\rho[\mathsf{an}(C_i), \mathsf{an}(C'_i)]\big)\ \mathsf{out}\big(\mathsf{tr}(C'_i)\big)$$

has period $p \leqslant \boldsymbol{B}$ dividing $|\mathsf{out}(\mathsf{tr}(C_i))|$ and $|\mathsf{out}(\mathsf{tr}(C'_i))|$.

Similar to the characterization of k-pass sweeping definability in [10], we show now the following characterization for 2-way transducers, using Theorem 24 as a black-box:

Theorem 35. *Let $\mathcal{T}$ be a functional two-way transducer and $k > 0$. The following are equivalent:*

1) *$\mathcal{T}$ is k-pass sweeping definable.*
2) *All k-inversions of all successful runs of $\mathcal{T}$ are safe.*

The problem of deciding whether the above conditions hold is in 2EXPSPACE*; more precisely, it can be decided in double exponential space w.r.t. $|\mathcal{T}|$ and in polynomial space w.r.t. k.*

Proof sketch. A proof of this result (modulo the necessary changes in complexity due to the new characterization) can be found in [10]. Here we present in an informal way the main steps of the proof.

Proving the implication from 2) to 1) boils down to factorize a successful run ρ of $\mathcal{T}$ into factors $\rho_1, \dots, \rho_k$ in such a way that, for every odd (resp. even) index i, ρ_i contains only inversions (resp. co-inversions) that are safe, namely, that yield periodic outputs. We use the constructions presented in Section VI to simulate the output of each factor ρ_i with a one-way transducer, which scans the input either from left to right or from right to left, depending on whether i is odd or even.

The implication from 1) to 2) amounts at showing that every k-inversion is safe under the assumption that $\mathcal{T}$ is k-pass sweeping definable. The proof builds upon the characterization of one-way definability. More precisely, we consider a successful run of $\mathcal{T}$ and the corresponding run of an equivalent k-pass sweeping transducer $\mathcal{T}'$ that produces the same output. We then pump those runs simultaneously on all loops $L_1, \dots, L_{2k}$ that form the k-inversion. By reasoning as in the proof of Proposition 16, we derive a periodicity property that shows that the k-inversion is safe.

Finally, the 2EXPSPACE complexity of the decision problem follows from reducing k-pass sweeping definability to the emptiness of the language $\mathsf{dom}(\mathcal{T}) \cap U^{\complement}$, where U is now the language of words $u \in \mathsf{dom}(\mathcal{T})$ such that all k-inversions of all successful runs on u are safe. As usual the latter problem is solved by constructing an NFA that recognizes $U^{\complement}$ by guessing a successful run ρ of $\mathcal{T}$ and an unsafe k-inversion of ρ. $\square$

A similar problem, called *sweeping definability*, concerns the characterization of those transductions that are definable by sweeping transducers, but this time without enforcing any bound on the number of passes (or reversals). Of course the latter problem is interesting only when the transductions are presented by means of two-way transducers. Below we show that the sweeping definability problem reduces to the k-pass sweeping definability problem, when we set k large enough.

Theorem 36. *A functional two-way transducer $\mathcal{T}$ is sweeping definable iff it is k-pass sweeping definable, for $k = 2h_{\mathsf{max}} \cdot (2^{3e_{\mathsf{max}}} + 1)$.*

Proof sketch. The right-to-left implication is trivial. The proof of the converse direction is in the appendix; here we only provide a rough idea. Suppose that $\mathcal{T}$ is not k-pass sweeping definable, for $k = 2h_{\mathsf{max}} \cdot (2^{3e_{\mathsf{max}}} + 1)$. By Theorem 35, there exists a successful run ρ of $\mathcal{T}$ and an unsafe k-inversion $\overline{\mathcal{I}}$ of ρ. One can exploit the fact that k is large enough to find an idempotent loop L and an intercepted factor of it that covers two consecutive (co-)inversions of $\overline{\mathcal{I}}$. Then, by pumping the loop L, one can introduce arbitrarily long alternations between inversions and co-inversions, thus showing that there are successful runs with unsafe k'-inversions for all $k' > 0$. By Theorem 35, this proves that $\mathcal{T}$ is not sweeping definable. $\square$

Corollary 37. *The problem of deciding sweeping definability of a functional two-way transducer is in* 2EXPSPACE.

Another consequence is that it is decidable in 2EXPSPACE whether a functional two-way transducer is equivalent to some two-way transducer performing a bounded number of reversals in every run. Indeed, in [10] we proved that a functional transducer is k-pass sweeping definable iff it is $(k-1)$-reversal definable.

Other classes of transducers are amenable to characterizations via similar techniques. For example, we may consider an even more restricted variant of transducer, called *rotating transducer*. This is a sweeping transducer that emits output only when moving from left to right. Such a transducer is called *k-pass* if it performs at most k passes from left to right. To characterize those transductions that are definable by k-pass rotating transducers it suffices to modify slightly the definition of k-inversion, by removing co-inversions. Formally, one defines a *rotating k-inversion* as a tuple $\overline{\mathcal{I}} = (\mathcal{I}_0, \dots, \mathcal{I}_{k-1})$, where each $\mathcal{I}_i = (L_i, C_i, L'_i, C'_i)$ is an inversion and $\mathsf{an}(C'_i) \lhd \mathsf{an}(C_{i+1})$ for all $i < k - 1$. The analogous of Theorems 35 and 36 would then carry over.

VIII. CONCLUSIONS

It was shown recently [8] that it is decidable whether a given two-way transducer can be implemented by some one-way transducer, however the complexity of the algorithm is non-elementary.

The main contribution of our paper is a new algorithm that solves the above question with elementary complexity, precisely in 2EXPSPACE. The algorithm is based on a characterization of those transductions, given as two-way transducers, that can be realized by one-way transducers. The flavor of our characterization is different from that of [8]. The approach from [8] is based on a variant of Rabin and Scott's construction [4] of one-way automata, and on local modifications of the two-way run. Our approach relies instead on the global notion of *inversions* and on combinatorial arguments, and is inspired by our previous result for sweeping transducers [9]. The technical challenge in this paper compared to [9] is however significant, and required several involved proof ingredients, ranging from the type of loops we consider, up to the decomposition of the runs.

Our characterization based on inversions yields not only an elementary solution for the problem of one-way definability, but also for definability by sweeping (resp. rotating) transducers, with either known or unknown number of passes. All characterizations above are effective, and can be decided in 2EXPSPACE.

References

[1] M. Schützenberger, "A remark on finite transducers," *Information and Control*, vol. 4, no. 2-3, pp. 185–196, 1961.

[2] A. Aho, J. Hopcroft, and J. Ullman, "A general theory of translation," *Math. Syst. Theory*, vol. 3, no. 3, pp. 193–221, 1969.

[3] S. Eilenberg, *Automata, Langages and Machines*. Academic Press, 1976.

[4] M. Rabin and D. Scott, "Finite automata and their decision problems," *IBM J. Res. Dev.*, vol. 3, no. 2, pp. 114–125, 1959.

[5] J. Shepherdson, "The reduction of two-way automata to one-way automata," *IBM J. Res. Dev.*, vol. 3, no. 2, pp. 198–200, 1959.

[6] J. Engelfriet and H. J. Hoogeboom, "MSO definable string transductions and two-way finite-state transducers," *ACM Trans. Comput. Logic*, vol. 2, no. 2, pp. 216–254, 2001.

[7] R. Alur and P. Cerný, "Expressiveness of streaming string transducers." in *FSTTCS*, ser. LIPIcs, vol. 8. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010, pp. 1–12.

[8] E. Filiot, O. Gauwin, P. Reynier, and F. Servais, "From two-way to one-way finite state transducers," in *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*. IEEE Computer Society, 2013, pp. 468–477.

[9] F. Baschenis, O. Gauwin, A. Muscholl, and G. Puppis, "One-way definability of sweeping transducer," in *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, December 16-18, 2015, Bangalore, India*, ser. LIPIcs, vol. 45. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, pp. 178–191.

[10] ——, "Minimizing resources of sweeping and streaming string transducers," in *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, ser. LIPIcs, vol. 55. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016, pp. 114:1–114:14, full version available at https://hal.archives-ouvertes.fr/hal-01274992.

[11] E. Filiot, S. N. Krishna, and A. Trivedi, "First-order definable string transformations," in *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India*, ser. LIPIcs, vol. 29. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014, pp. 147–159.

[12] O. Carton and L. Dartois, "Aperiodic two-way transducers and FO-transductions," in *24th EACSL Annual Conference on Computer Science Logic (CSL)*, ser. LIPIcs, vol. 41. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015, pp. 160–174.

[13] E. Filiot, O. Gauwin, and N. Lhote, "First-order definability of rational transductions: An algebraic approach," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS'16, New York, NY, USA, July 5-8, 2016*. ACM, 2016, pp. 387–396.

[14] L. Daviaud, P. Reynier, and J. Talbot, "A generalised twinning property for minimisation of cost register automata," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS'16, New York, NY, USA, July 5-8, 2016*. ACM, 2016, pp. 857–866.

[15] C. Choffrut and B. Guillon, "An algebraic characterization of unary two-way transducers," in *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 8634. Springer, 2014, pp. 196–207.

[16] B. Guillon, "Sweeping weakens two-way transducers even with a unary output alphabet," in *Seventh Workshop on Non-Classical Models of Automata and Applications - NCMA 2015, Porto, Portugal, August 31 - September 1, 2015. Proceedings*, ser. books@ocg.at, vol. 318. Österreichische Computer Gesellschaft, 2015, pp. 91–108.

[17] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[18] J. Birget, "Two-way automaton computations," *RAIRO - Theoretical Informatics and Applications - Informatique Thorique et Applications*, vol. 24, no. 1, pp. 47–66, 1990.

[19] I. Simon, "Factorization forests of finite height," *Theoretical Computer Science*, vol. 72, no. 1, pp. 65–94, 1990.

[20] T. Colcombet, "Factorisation forests for infinite words," in *FCT*, ser. LNCS, vol. 4639. Springer, 2007, pp. 226–237.

[21] N. Fine and H. Wilf, "Uniqueness theorems for periodic functions," *Proceedings of the American Mathematical Society*, vol. 16, pp. 109–114, 1965.

[22] J. Kortelainen, "On the system of word equations $x_0u_1^ix_1u_2^ix_2\dots u_m^ix_m = y_0v_1^iy_1v_2^iy_2\dots v_m^iy_m$ $(i = 0, 1, 2, \dots)$ in a free monoid," *Journal of Automata, Languages and Combinatorics*, vol. 3, no. 1, pp. 43–57, 1998.

[23] A. Saarela, "Systems of word equations, polynomials and linear algebra: a new approach," *European Journal of Combinatorics*, vol. 47, no. 5, pp. 1–14, 2015.

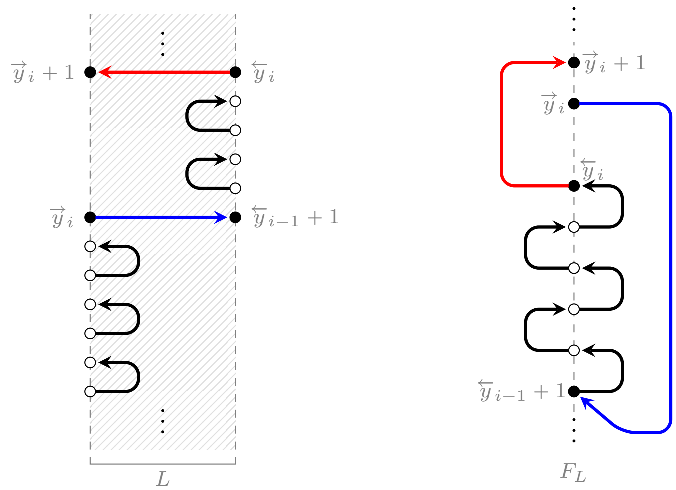


Fig. 8. Some factors intercepted by L and the corresponding edges in the flow.

Appendix

Before proving Lemma 8, we show that in a loop, the levels of each component form an interval.

Lemma 38. *Let C be a component of a loop $L = [x_1, x_2]$, $y^- = \min(C)$, and $y^+ = \max(C)$. The nodes of C are precisely the levels in the interval $[y^-, y^+]$. Moreover, if C is left-to-right (resp. right-to-left), then y^+ is the smallest level $\geqslant y^-$ such that between (x_1, y^-) and (x_2, y^+) (resp. (x_2, y^-) and (x_1, y^+)) there are equally many* LL*-factors and* RR*-factors intercepted by L.*

Proof. To ease the understanding the reader may refer to Fig. 8, that shows some factors intercepted by L and the corresponding edges in the flow.

We begin the proof by partitioning the set of levels of the flow into suitable intervals as follows. We observe that every loop $L = [x_1, x_2]$ intercepts equally many LL-factors and RR-factors. This is so because the crossing sequences at x_1, x_2 have the same length h. We also observe that the sources of the factors intercepted by L are either of the form (x_1, y), with y even, or (x_2, y), with y odd. For any location $\ell \in \{x_1, x_2\} \times \mathbb{N}$ that is the source of an intercepted factor, we define d_ℓ to be the difference between the number of LL-factors and the number of RR-factors intercepted by L that *end* at a location *strictly before* ℓ. Intuitively, $d_\ell = 0$ when the prefix of the run up to location ℓ has visited equally many times the position x_1 and the position x_2. For the sake of brevity, we let $d_y = d_{(x_1,y)}$ for an even level y, and $d_y = d_{(x_2,y)}$ for an odd level y. Note that $d_0 = 0$. We also let $d_{h+1} = 0$, by convention.

We now consider the numbers z's, with $0 \leqslant z \leqslant h+1$, such that $d_z = 0$, that is: $0 = z_0 < z_1 < \cdots < z_k = h+1$. Using a simple induction, we prove that for all $i \leqslant k$, the parity of z_i is the same as the parity of its index i. The base case $i = 0$ is trivial, since $z_0 = 0$. For the inductive case, suppose that z_i is even (the case of z_i odd is similar). We prove that z_{i+1} is odd by a case distinction based on the type of factor intercepted by L that starts at level z_i. If this factor is an LR-factor, then it ends at the same level z_i, and hence $d_{z_i+1} = d_{z_i} = 0$, which implies that $z_{i+1} = z_i + 1$ is odd. Otherwise, if the factor is an LL-factor, then for all levels z strictly between z_i and z_{i+1}, we have $d_z > 0$, and since $d_{z_{i+1}} = 0$, the last factor before z_{i+1} must decrease d_z, that is, must be an RR-factor. This implies that (x_2, z_{i+1}) is the source of an intercepted factor, and thus z_{i+1} is odd.

The levels $0 = z_0 < z_1 < \cdots < z_k = h+1$ induce a partition of the set of nodes of the flow into intervals of the form $Z_i = [z_i, z_{i+1} - 1]$. To prove the lemma, it is suffices to show that the subgraph of the flow induced by each interval Z_i is connected. Indeed, because the union of the previous intervals covers all the nodes of the flow, and because each node has one incoming and one outgoing edge, this will imply that the intervals coincide with the components of the flow.

Now, let us fix an interval of the partition, which we denote by Z to avoid clumsy notation. Hereafter, we will focus on the edges of subgraph of the flow induced by Z (we call it *subgraph of Z* for short). We prove a few basic properties of these edges. For the sake of brevity, we call LL-edges the edges of the subgraph of Z that correspond to the LL-factors intercepted by L, and similarly for the RR-edges, LR-edges, and RL-edges.

We make a series of assumption to simplify our reasoning. First, we assume that the edges are ordered based on the occurrences of the corresponding factors in the run. For instance, we may say the first, second, etc. LR-edge (of the subgraph of Z) — from now on, we tacitly assume that the edges are inside the subgraph of Z. Second, we assume that the first edge of the subgraph of Z starts at an even node, namely, it is an LL-edge or an LR-edge (if this were not the case, one could apply symmetric arguments to prove the lemma). From this it follows that the subgraph contains n LR-edges interleaved by $n-1$ RL-edges, for some $n > 0$. Third, we assume that $\min(Z) = 0$, in order to avoid clumsy notations (otherwise, we need to add $\min(Z)$ to all the levels considered hereafter).

Now, we observe that, by definition of Z, there are equally many LL-edges and RR-edges: indeed, the difference between the number of LL-edges and the number of RR-edges at the beginning and at the end of Z is the same, namely, $d_z = 0$ for both $z = \min(Z)$ and $z = \max(Z)$. It is also easy to see that the LL-edges and the RR-edges are all of the form $y \to y+1$, for some level y. We call these edges *incremental edges*.

For the other edges, we denote by $\overrightarrow{y}_i$ (resp. $\overleftarrow{y}_i$) the source level of the i-th LR-edge (resp. the i-th RL-edge). Clearly, each $\overrightarrow{y}_i$ is even, and each $\overleftarrow{y}_i$ is odd, and $i \leqslant j$ implies $\overrightarrow{y}_i < \overrightarrow{y}_j$ and $\overleftarrow{y}_i < \overleftarrow{y}_j$. Consider the location $(x_1, \overrightarrow{y}_i)$, which is the source of the i-th LR-edge (e.g. the edge in blue in the figure). The latest location at position x_2 that precedes $(x_1, \overrightarrow{y}_i)$ must be of the form $(x_2, \overleftarrow{y}_{i-1})$, provided that $i > 1$. This implies that, for all $1 < i \leqslant n$, the i-th LR-edge is of the form $\overrightarrow{y}_i \to \overleftarrow{y}_{i-1} + 1$. For $i = 1$, we recall that $\min(Z) = 0$ and observe that the first location at position x_2 that occurs after the location $(x_1, 0)$ is $(x_2, 0)$, and thus the first LR-edge has a similar form: $\overrightarrow{y}_1 \to \overleftarrow{y}_0 + 1$, where $\overleftarrow{y}_0 = -1$ by convention.

Using symmetric arguments, we see that the i-th RL-edge (e.g. the one in red in the figure) is of the form $\overleftarrow{y}_i \to \overrightarrow{y}_i + 1$. In particular, the last LR-edge starts at the level $\overrightarrow{y}_n = \max(Z)$.

Summing up, we have just seen that the edges of the subgraph of Z are of the following forms:

- $y \to y+1$ (incremental edges),
- $\overrightarrow{y}_i \to \overleftarrow{y}_{i-1} + 1$ (i-th LR-edge, for $i = 1, \ldots, n$),
- $\overleftarrow{y}_i \to \overrightarrow{y}_i + 1$ (i-th RL-edge, for $i = 1, \ldots, n-1$).

In addition, we have $\overrightarrow{y}_i + 1 = \overleftarrow{y}_i + 2d_{\overleftarrow{y}_i}$. Since $d_z > 0$ for all $\min(Z) < z < \max(Z)$, this implies that $\overrightarrow{y}_i > \overleftarrow{y}_i$.

The goal is to prove that the subgraph of Z is strongly connected, namely, it contains a cycle that visits all its nodes. As a matter of fact, because components are also strongly connected subgraphs, and because every node in the flow has in-/out-degree 1, this will imply that the considered subgraph coincides with a component C, thus implying that the nodes in C form an interval. Towards this goal, we will prove a series of claims that aim at identifying suitable sets of nodes that are covered by paths in the subgraph of Z. Formally, we say that a path *covers* a set Y if it visits all the nodes in Y, and possibly other nodes. As usual, when we talk of edges or paths, we tacitly understand that they occur inside the subgraph of Z. On the other hand, we do not need to assume $Y \subseteq Z$, since this would follow from the fact that Y is covered by a path inside Z. For example, the right hand-side of Fig. 8 shows a path from $\overrightarrow{y}_i$ to $\overrightarrow{y}_i + 1$ that covers the set $Y = \{\overrightarrow{y}_i, \overrightarrow{y}_i + 1\} \cup [\overleftarrow{y}_{i-1} + 1, \overleftarrow{y}_i]$.

The covered sets will be intervals of the form

$$Y_i \ = \ [\overleftarrow{y}_{i-1} + 1, \overleftarrow{y}_i].$$

Note that the sets Y_i are well-defined for all $i = 1, \ldots, n-1$, but not for $i = n$ since $\overleftarrow{y}_n$ is not defined either (the subgraph of Z contains only $n-1$ RL-edges).

Claim. *For all $i = 1, \ldots, n-1$, there is a path from $\overrightarrow{y}_i$ to $\overrightarrow{y}_i + 1$ that covers Y_i (for short, we call it an* incremental path*).*

Proof. We prove the claim by induction on i. The base case $i = 1$ is rather easy. Indeed, we recall the convention that $\overleftarrow{y}_0 + 1 = \min(Z) = 0$. In particular, the node $\overleftarrow{y}_0 + 1$ is the target of the first LR-edge of the subgraph of Z. Before this edge, according to the order induced by the run, we can only have LL-edges of the form $y \to y+1$, with $y = 0, 2, \ldots, \overrightarrow{y}_1 - 2$. Similarly, after the LR-edge we have RR-edges of the form $y \to y+1$, with $y = 1, 3, \ldots, \overleftarrow{y}_1 - 2$. Those incremental edges can be connected to form the path $\overleftarrow{y}_{i-1} + 1 \to^* \overleftarrow{y}_1$ that covers the interval $[\overleftarrow{y}_0 + 1, \overleftarrow{y}_1]$. By prepending to this path the LR-edge $\overrightarrow{y}_1 \to \overleftarrow{y}_0 + 1$, and by appending the RL-edge $\overleftarrow{y}_1 \to \overrightarrow{y}_1 + 1$, we get a path from $\overrightarrow{y}_1$ to $\overrightarrow{y}_1 + 1$ that covers the interval $[\overleftarrow{y}_0 + 1, \overleftarrow{y}_1]$. The latter interval is precisely the set Y_1.

For the inductive step, we fix $1 < i < n$ and we construct the desired path from $\overrightarrow{y}_i$ to $\overrightarrow{y}_i + 1$. The initial edge of this path is defined to be the LR-edge $\overrightarrow{y}_i \to \overleftarrow{y}_{i-1} + 1$. Similarly, the final edge of the path will be the RL-edge $\overleftarrow{y}_i \to \overrightarrow{y}_i + 1$, which exists since $i < n$. It remains to connect $\overleftarrow{y}_{i-1} + 1$ to $\overleftarrow{y}_i$. For this, we consider the edges that depart from nodes strictly between $\overleftarrow{y}_{i-1}$ and $\overleftarrow{y}_i$.

Let y be an arbitrary node in $[\overleftarrow{y}_{i-1} + 1, \overleftarrow{y}_i - 1]$. Clearly, y cannot be of the form $\overleftarrow{y}_j$, for some j, because it is strictly between $\overleftarrow{y}_{i-1}$ and $\overleftarrow{y}_i$. So y cannot be the source of an RL-edge. Moreover, recall that the LL-edges and the RR-edges are the of the form $y \to y+1$. As these incremental edges do not pose particular problems for the construction of the path, we focus mainly on the LR-edges that depart from nodes inside $[\overleftarrow{y}_{i-1} + 1, \overleftarrow{y}_i - 1]$.

Let $\overrightarrow{y}_j \to \overleftarrow{y}_{j-1} + 1$ be such an LR-edge, for some j such that $\overrightarrow{y}_j \in [\overleftarrow{y}_{i-1} + 1, \overleftarrow{y}_i - 1]$. If we had $j \geqslant i$, then we would have $\overrightarrow{y}_j \geqslant \overrightarrow{y}_i > \overleftarrow{y}_i$, but this would contradict the assumption that $\overrightarrow{y}_j \in [\overleftarrow{y}_{i-1} + 1, \overleftarrow{y}_i - 1]$. So we know that $j < i$. This enables the use of the inductive hypothesis, which implies the existence of an incremental path from $\overrightarrow{y}_j$ to $\overrightarrow{y}_j + 1$ that covers the interval Y_j.

Finally, by connecting the above paths using the incremental edges, and by adding the initial and final edges $\overrightarrow{y}_i \to \overleftarrow{y}_{i-1}+1$ and $\overleftarrow{y}_i \to \overrightarrow{y}_i + 1$, we obtain a path from $\overrightarrow{y}_i$ to $\overrightarrow{y}_i + 1$. It is easy to see that this path covers the interval Y_i. ▫ *(claim)*

Next, we define

$$Y \;=\; [\overleftarrow{y}_{n-1}+1, \overrightarrow{y}_n] \;\cup \bigcup_{1\leqslant i<n} Y_i.$$

We prove a claim similar to the previous one, but now aiming to cover Y with a *cycle*. Towards the end of the proof we will argue that the set Y coincides with the full interval Z, thus showing that there is a component C whose set of notes is precisely Z.

Claim. *There is a cycle that covers Y.*

Proof. It is convenient to construct our cycle starting from the last LR-edge, that is, $\overrightarrow{y}_n \to \overleftarrow{y}_{n-1}+1$, since this will cover the upper node $\overrightarrow{y}_n = \max(Z)$. From there we continue to add edges and incremental paths, following an approach similar to the proof of the previous claim, until we reach the node $\overrightarrow{y}_n$ again. More precisely, we consider the edges that depart from nodes strictly between $\overleftarrow{y}_{n-1}$ and $\overrightarrow{y}_n$. As there are only $n-1$ RL-edges, we know that every node in the interval $[\overleftarrow{y}_{n-1}+1, \overrightarrow{y}_n - 1]$ must be source of an LL-edge, an RR-edge, or an LR-edge. As usual, incremental edges do not pose particular problems for the construction of the cycle, so we focus on the LR-edges. Let $\overrightarrow{y}_i \to \overleftarrow{y}_{i-1}+1$ be such an LR-edge, with $\overrightarrow{y}_i \in [\overleftarrow{y}_{n-1}+1, \overrightarrow{y}_n - 1]$. Since $i < n$, we know from the previous claim that there is a path from $\overrightarrow{y}_i$ to $\overrightarrow{y}_i + 1$ that covers Y_i. We can thus build a cycle π by connecting the above paths using the incremental edges and the LR-edge $\overrightarrow{y}_n \to \overleftarrow{y}_{n-1}+1$.

By construction, the cycle π covers the interval $[\overleftarrow{y}_{n-1}+1, \overrightarrow{y}_n]$, and for every $i < n$, if π visits $\overrightarrow{y}_i$, then π covers Y_i. So to complete the proof — namely, to show that π covers the entire set Y — it suffices to prove that π visits each node $\overrightarrow{y}_i$, with $i < n$.

Suppose, by way of contradiction, that $\overrightarrow{y}_i$ is the node with the highest index $i < n$ that is not visited by π. Recall that $\overrightarrow{y}_i > \overleftarrow{y}_i$. This shows that

$$\overrightarrow{y}_i \;\in\; [\overleftarrow{y}_i+1, \overrightarrow{y}_n] \;=\; \bigcup_{i\leqslant j<n-1} [\overleftarrow{y}_j+1, \overleftarrow{y}_{j+1}] \;\cup\; [\overleftarrow{y}_{n-1}+1, \overrightarrow{y}_n].$$

As we already proved that π covers the interval $[\overleftarrow{y}_{n-1}+1, \overrightarrow{y}_n]$, we know that $\overrightarrow{y}_i \in [\overleftarrow{y}_j+1, \overleftarrow{y}_{j+1}]$ for some j with $i \leqslant j < n-1$. Now recall that $\overrightarrow{y}_i$ is the highest node that is not visited by π. This means that $\overrightarrow{y}_{j+1}$ is visited by π. Moreover, since $j+1 < n$, we know that π uses the incremental path from $\overrightarrow{y}_{j+1}$ to $\overrightarrow{y}_{j+1}+1$, which covers $Y_{j+1} = [\overleftarrow{y}_j+1, \overleftarrow{y}_{j+1}]$. But this contradicts the fact that $\overrightarrow{y}_i$ is not visited by π, since $\overrightarrow{y}_i \in [\overleftarrow{y}_j+1, \overleftarrow{y}_{j+1}]$. ▫ *(claim)*

We know that the set Y is covered by a cycle of the subgraph of Z, and that Z is an interval whose endpoints are consecutive levels $z < z'$, with $d_z = d_{z'} = 0$. For the homestretch, we prove that $Y = Z$. This will imply that the nodes of the cycle are precisely the nodes of the interval Z. Moreover, because the cycle must coincide with a component C of the flow (recall that all the nodes have in-/out-degree 1), this will show that the nodes of C are precisely those of Z.

To prove $Y = Z$ it suffices to recall its definition as the union of the interval $[\overleftarrow{y}_{n-1}+1, \overrightarrow{y}_n]$ with the sets Y_i, for all $i = 1, \dots, n-1$. Clearly, we have that $Y \subseteq Z$. For the converse inclusion, we also recall that $\overleftarrow{y}_0 + 1 = 0 = \min(Z)$ and $\overrightarrow{y}_n = \max(Z)$. Consider an arbitrary level $z \in Z$. Clearly, we have either $z \leqslant \overleftarrow{y}_i$, for some $1 \leqslant i < n$, or $z > \overleftarrow{y}_n$. In the former case, by choosing the smallest index i such that $z \leqslant \overleftarrow{y}_i$, we get $z \in [\overleftarrow{y}_{i-1}+1, \overleftarrow{y}_i]$, whence $z \in Y_i \subseteq Y$. In the latter case, we immediately have $z \in Y$, by construction. ☐

Lemma 8. *If C is a left-to-right (resp. right-to-left) component of an* idempotent *loop L, then the (L,C)-factors are in the following order: k LL-factors (resp. RR-factors), followed by one LR-factor (resp. RL-factor), followed by k RR-factors (resp. LL-factors), for some $k \geqslant 0$.*

Proof. Suppose that C is a left-to-right component of L. We show by way of contradiction that C has only one LR-factor and no RL-factor. By Lemma 38 this will yield to the claimed shape. Fig. 9 can be used as a reference example for the arguments that follow.

We begin by listing the (L,C)-factors. As usual, we order them based on their occurrences in the run ρ. Let γ be the first (L,C)-factor that is not an LL-factor, and let $\beta_1, \dots, \beta_k$ be the (L,C)-factors that precede γ (these are all LL-factors). Because γ starts at an even level, it must be an LR-factor. Suppose that there is another (L,C)-factor, say ζ, that comes after γ and it is neither an RR-factor nor an LL-factor. Because ζ starts at an odd level, it must be an RL-factor. Further let $\delta_1, \dots, \delta_{k'}$ be the intercepted RR-factors that occur between γ and ζ. We claim that $k' < k$, namely, that the number of RR-factors between γ and ζ is strictly less than the number of LL-factors before γ. Indeed, if this were not the case, then, by Lemma 38, the level where ζ starts would not belong to the component C.

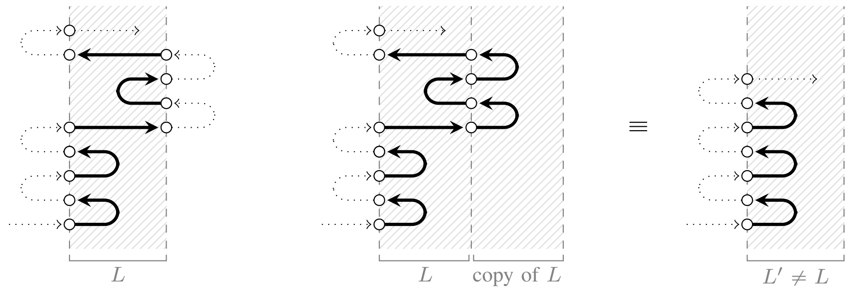


Fig. 9. Pumping a loop L with a wrong shape and showing it is not idempotent.

Now, consider the pumped run $\rho' = \mathsf{pump}^2_L(\rho)$, obtained by adding a new copy of L. Let L' be the loop of ρ' obtained from the union of L and its copy. Since L is idempotent, the components of L are isomorphic to the components of L'. In particular, we can denote by C' the component of L' that is isomorphic to C. Let us consider the (L', C')-factors of ρ'. The first k such factor are isomorphic to the k LL-factors $\beta_1, \ldots, \beta_k$ from ρ. However, the $(k+1)$-th element has a different shape: it is isomorphic to $\gamma\ \beta_1\ \delta_1\ \beta_2\ \ldots\ \delta_{k'}\ \beta_{k'+1}\ \zeta$, and in particular it is an LL-factor. This implies that the $(k+1)$-th edge of C' is of the form $(y, y+1)$, while the $(k+1)$-th edge of C is of the form $(y, y-2k)$. This contradiction comes from having assumed the existence of the RL-factor ζ, and is illustrated in Fig. 9. ☐

The following lemma will be used to prove Theorem 13.

Lemma 39. *If $L_1 = [x_1, x_2]$ and $L_2 = [x_2, x_3]$ are consecutive idempotent loops with the same effect and α, β are two factors intercepted by L_1, L_2 that are adjacent in the run (namely, they share the endpoint at position x_2), then α and β correspond to edges of the same component of L_1 (or, equally, L_2).*

Proof. Let C be the component of L_1 and (y, y') the edge of C that corresponds to the factor α intercepted by L_1. Similarly, let C' be the component of L_2 and (y'', y''') the edge of C' that corresponds to the factor β intercepted by L_2. Since α and β share the endpoint at position x_2, we know that $y' = y''$. This shows that $C \cap C' \neq \varnothing$, and hence $C = C'$. ☐

Proposition 11. *Let L be an idempotent loop of ρ with components $C_1, \ldots, C_k$, listed according to the order of their anchors: $\mathsf{an}(C_1) \lhd \cdots \lhd \mathsf{an}(C_k)$. For all $m \in \mathbb{N}$, we have*

$$\mathsf{pump}^{m+1}_L(\rho) \ = \ \rho_0\ \mathsf{tr}(C_1)^m\ \rho_1\ \cdots\ \rho_{k-1}\ \mathsf{tr}(C_k)^m\ \rho_k$$

where

- *ρ_0 is the prefix of ρ that ends at $\mathsf{an}(C_1)$,*
- *ρ_i is the factor of ρ between $\mathsf{an}(C_i)$ and $\mathsf{an}(C_{i+1})$, for all $i = 1, \ldots, k-1$,*
- *ρ_k is the suffix of ρ that starts at $\mathsf{an}(C_k)$.*

Proof. Along the proof we sometimes refer to Fig. 4 to ease the intuition of some definitions and arguments. Let $L = [x_1, x_2]$ be an idempotent loop and, for all $i = 0, \ldots, m$, let $L'_i = [x'_i, x'_{i+1}]$ be the i-th copy of the loop L in the pumped run $\rho' = \mathsf{pump}^{m+1}_L(\rho)$, where $x'_i = x_1 + i \cdot (x_2 - x_1)$ (the "0-th copy of L" is the loop L itself). Further let $L' = L'_0 \cup \cdots \cup L'_m = [x'_0, x'_{m+1}]$, that is, L' is the loop of ρ' that spans across the $m+1$ occurrences of L. As L is idempotent, the loops $L'_0, \ldots, L'_m$ and L' have all the same effect as L. In particular, the components of $L'_0, \ldots, L'_m$, and L' are isomorphic to and in same order as those of L. We denote these components by $C_1, \ldots, C_k$.

We let $\ell_j = \mathsf{an}(C_j)$ be the anchor of each component C_j inside the loop L of ρ (these locations are marked by black dots in the left hand-side of Fig. 4). Similarly, we let $\ell'_{i,j}$ (resp. ℓ'_j) be the anchor of C_j inside the loop L'_i (resp. L'). From Definition 9, we have that either $\ell'_j = \ell'_{1,j}$ or $\ell'_j = \ell'_{m,j}$, depending on whether C_j is left-to-right or right-to-left (or, equally, on whether j is odd or even).

Now, let us consider the factorization of the pumped run ρ' induced by the locations $\ell'_{i,j}$, for all $i = 0, \ldots, m$ and for $j = 1, \ldots, k$ (these locations are marked by black dots in the right hand-side of the figure). By construction, the prefix of ρ' that ends at location $\ell'_{0,1}$ coincides with the prefix of ρ that ends at ℓ_1, i.e. ρ_0 in the statement of the proposition. Similarly, the suffix of ρ' that starts at location $\ell'_{m,k}$ is isomorphic to the suffix of ρ that starts at ℓ_k, i.e. ρ_k in the statement. By construction, we also know that, for all odd (resp. even) indices j, the factor $\rho'[\ell'_{m,j}, \ell'_{m,j+1}]$ (resp. $\rho'[\ell_{0,j}, \ell_{0,j+1}]$) is isomorphic to $\rho[\ell_j, \ell_{j+1}]$, i.e. the ρ_j of the statement.

The remaining factors of ρ' are those delimited by the pairs of locations $\ell'_{i,j}$ and $\ell'_{i+1,j}$, for all $i = 0, \ldots, m-1$ and all $j = 1, \ldots, k$. Consider one such factor $\rho'[\ell'_{i,j}, \ell'_{i+1,j}]$, and assume that the index j is odd (the case of an even j is similar).

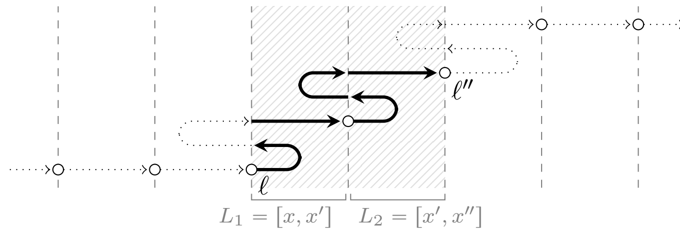


Fig. 10. Two consecutive idempotent loops with the same effect.

This factor can be seen as a concatenation of factors intercepted by L that correspond to edges of C_j inside L'_i. More precisely, $\rho'[\ell'_{i,j}, \ell'_{i+1,j}]$ is obtained by concatenating the unique LR-factor of C_j — recall that by Lemma 8 there is exactly one such factor — with an interleaving of the LL-factors and the RR-factors of C_j. As the components are the same for all L'_i's, this corresponds precisely to the trace $\mathsf{tr}(C_j)$ (cf. Definition 10). Now that we know that $\rho'[\ell'_{i,j}, \ell'_{i+1,j}]$ is isomorphic to $\mathsf{tr}(C_j)$, we can conclude that $\rho'[\ell'_{0,j}, \ell'_{m,j}] = \rho'[\ell'_{0,j}, \ell'_{1,j}] \ \dots\ \rho'[\ell'_{m-1,j}, \ell'_{m,j}]$ is isomorphic to $\mathsf{tr}(C_j)^m$. ☐

Theorem 13. *Let $I = [x_1, x_2]$ be an interval of positions, $K = [\ell_1, \ell_2]$ an interval of locations, and $Z = K \cap (I \times \mathbb{N})$. If $\big|\mathsf{out}(\rho \mid Z)\big| > \boldsymbol{B}$, then there exist an idempotent loop L and a component C of L such that*

- $x_1 < \min(L) < \max(L) < x_2$ *(in particular, $L \subsetneq I$),*
- $\ell_1 \lhd \mathsf{an}(C) \lhd \ell_2$ *(in particular, $\mathsf{an}(C) \in K$),*
- $\mathsf{out}(\mathsf{tr}(C)) \neq \varepsilon$.

Proof. Let I, K, Z be as in the statement, and suppose that $\big|\mathsf{out}(\rho \mid Z)\big| > \boldsymbol{B}$. We define $Z' = Z \setminus \{\ell_1, \ell_2\} \setminus \big(\{x_1, x_2\}\big) \times \mathbb{N}\big)$ and we observe that there are at most $2h_{\mathsf{max}}$ locations in Z that are missing from Z'. This means that $\rho \mid Z'$ contains all but $4h_{\mathsf{max}}$ transitions of $\rho \mid Z$, and because each transition outputs at most c_{max} letters, we have $\big|\mathsf{out}(\rho \mid Z')\big| > \boldsymbol{B} - 4c_{\mathsf{max}} \cdot h_{\mathsf{max}} = c_{\mathsf{max}} \cdot h_{\mathsf{max}} \cdot 2^{3e_{\mathsf{max}}}$.

For every level y, let X_y be the set of positions x such that (x, y) is the source location of a transition of $\rho \mid Z'$ that produces non-empty output. For example, if we refer to Fig. 10, the vertical dashed lines represent the positions of X_y for a particular level y; accordingly, the circles in the figure represent the locations of the form (x, y), for $x \in X_y$. Since each transition outputs at most c_{max} letters, we have $\sum_y |X_y| > h_{\mathsf{max}} \cdot 2^{3e_{\mathsf{max}}}$. Moreover, since there are at most h_{max} levels, there is a level y (which we fix hereafter) such that $|X_y| > 2^{3e_{\mathsf{max}}}$.

Claim. *There are two consecutive loops $L_1 = [x, x']$ and $L_2 = [x', x'']$ such that $E_{L_1} = E_{L_2} = E_{L_1 \cup L_2}$ and with endpoints $x, x', x'' \in X_y$.*

Proof. By Theorem 12, there is a factorization forest for X of height at most $3e_{\mathsf{max}}$. Since $|X_y| > 2^{3e_{\mathsf{max}}}$, we know that this factorization forest contains an internal node $L' = [x'_1, x'_{k+1}]$ with $k > 2$ children, say $L_1 = [x'_1, x'_2]$, $\dots L_k = [x'_k, x'_{k+1}]$. By definition of factorization forest, the effects $E_{L'}$, E_{L_1}, $\dots$, E_{L_k} are all equal and idempotent. Moreover, as ρ is a valid run, the dummy element $\bot$ of the effect semigroup does not appear in the factorization forest. In particular, the effect $E_{L'} = E_{L_1} = \dots = E_{L_k}$ is a triple of the form $(F_{L'}, c_1, c_2)$, where $c_i = \rho|x_i$ is the crossing sequence at x'_i. Finally, since $E_{L'}$ is idempotent, we have that $c_1 = c_2$ and this is equal to the crossing sequences of ρ at the positions $x'_1, \dots, x'_{k+1}$. This shows that L_1, L_2 are idempotent loops. ▫ *(claim)*

Turning back to the proof of the theorem, we know that there are two consecutive idempotent loops $L_1 = [x, x']$ and $L_2 = [x', x'']$ with the same effect and with endpoints $x, x', x'' \in X_y \subseteq I \setminus \{x_1, x_2\}$ (see again Fig. 10).

Let $\ell = (x, y)$ and $\ell'' = (x'', y)$, and observe that both locations belong to Z'. In particular, ℓ and ℓ'' are strictly between ℓ_1 and ℓ_2. Suppose by symmetry that $\ell \trianglelefteq \ell''$. Further let C be the component of $L_1 \cup L_2$ (or, equally, of L_1 or L_2) that contains the node y. Below, we focus on the factors of $\rho[\ell, \ell'']$ that are intercepted by $L_1 \cup L_2$: these are represented in Fig. 10 by the thick arrows. By Lemma 8 all these factors correspond to edges of the same component C, namely, they are $(L_1 \cup L_2, C)$-factors.

Consider any factor α of $\rho[\ell, \ell'']$ intercepted by $L_1 \cup L_2$, and assume that $\alpha = \beta_1 \cdots \beta_k$, where $\beta_1, \dots, \beta_k$ are the factors intercepted by L_1 or L_2. By Lemma 39, any two adjacent factors β_i, β_{i+1} correspond to edges in the same component of L_1 and L_2, respectively. Thus, by transitivity, all factors $\beta_1, \dots, \beta_k$ correspond to edges in the same component, say C'. We claim that $C' = C$. Indeed, if β_1 is intercepted by L_1, then $C' = C$ because α and β_1 start from the same location and hence they correspond to edges of the flow that depart from the same node. The other case is where β_1 is intercepted by L_2, for which a symmetric argument can be applied.

So far we have shown that every factor of $\rho[\ell,\ell']$ intercepted by $L_1 \cup L_2$ can be factorized into some (L_1,C)-factors and some (L_2,C)-factors. We conclude the proof with the following observations:

- By construction, both loops L_1, L_2 are contained in the interval of positions $I = [x_1,x_2]$, and have endpoints different from x_1, x_2.
- Both anchors of C inside L_1 and L_2 belong to the interval of locations $K \setminus \{\ell_1,\ell_2\}$. This holds because $\rho[\ell,\ell']$ contains a factor α that is intercepted by $L_1 \cup L_2$ and spans across all the positions from x to x'', namely, an LR-factor. This factor starts at the anchor of C inside L_1 and visits the anchor of C inside L_2. Moreover, by construction, α is also a factor of the subsequence $\rho \mid Z'$. This shows that the anchors of C inside L_1 and L_2 belong to Z', and in particular to $K \setminus \{\ell_1,\ell_2\}$.
- The first factor of $\rho[\ell,\ell']$ that is intercepted by $L_1 \cup L_2$ starts at $\ell = (x,y)$, which by construction is the source location of some transition producing non-empty output. By the previous arguments, this factor is a concatenation of (L_1,C)-factors and (L_2,C)-factors. This implies that the trace of C inside L_1 or the trace of C inside L_2 produces non-empty output. $\square$

Proposition 16. *If $\mathcal{T}$ is one-way definable, then for every inversion (L_1,C_1,L_2,C_2) of a successful run ρ of $\mathcal{T}$, the word*

$$\mathsf{out}\big(\mathsf{tr}(C_1)\big)\ \mathsf{out}\big(\rho[\mathsf{an}(C_1),\mathsf{an}(C_2)]\big)\ \mathsf{out}\big(\mathsf{tr}(C_2)\big)$$

has period p that divides both $|\mathsf{out}(\mathsf{tr}(C_1))|$ and $|\mathsf{out}(\mathsf{tr}(C_2))|$. Moreover, $p \leqslant \boldsymbol{B}$.

Proof of Proposition 16. The proof of the first claim of the proposition is similar to the proof of Proposition 7 in [9] for sweeping transducers. The main difficulty in the present proof is to get a bound on the period of the output of the inversion.

Let (L_1,C_1,L_2,C_2) be an inversion of a successful run ρ on input u. Note that the two loops L_1 and L_2 might not be disjoint. In fact, two cases arise: either $\max(L_2) < \min(L_1)$ (that is, L_1 and L_2 are disjoint and L_2 is strictly to the left of L_1), or $\min(L_1) \leqslant \min(L_2) \leqslant \max(L_1) \leqslant \max(L_2)$ (the fact that $\min(L_2) \leqslant \max(L_1)$ follows from the fact that the anchor $\mathsf{an}(C_2)$ is to the left of the anchor $\mathsf{an}(C_1)$). For the sake of simplicity, we only deal with the case where L_1 and L_2 are disjoint, as shown in Fig. 5 — the other case can be treated in a similar way by considering the rightmost copy of L_1 in the pumped run $\mathsf{pump}^3_{L_1}(\rho)$, which is clearly disjoint from the leftmost copy of L_2.

We begin by pumping the run ρ, together with the underlying input u, on the loops L_1 and L_2. Formally, for all numbers $m_1, m_2 \in \mathbb{N}$, we define

$$\begin{aligned} u^{(m_1,m_2)} &= \mathsf{pump}^{m_1+1}_{L_1}(\mathsf{pump}^{m_2}_{L_2}(u)) \\ \rho^{(m_1,m_2)} &= \mathsf{pump}^{m_1+1}_{L_1}(\mathsf{pump}^{m_2}_{L_2}(\rho)). \end{aligned}$$

We identify the positions that mark the endpoints of the occurrences of L_1 and L_2 in the pumped run $\rho^{(m_1,m_2)}$. Formally, if $L_1 = [x_1,x_2]$ and $L_2 = [x_3,x_4]$, then the sets of positions are defined as follows:

$$\begin{aligned} X_2^{(m_1,m_2)} &= \big\{x_3 + i\cdot(x_4-x_3) \ :\ i = 0,\dots,m_2+1\big\} \\ X_1^{(m_1,m_2)} &= \big\{x_1 + i\cdot(x_2-x_1) + m_2\cdot(x_4-x_3) \ :\ i = 0,\dots,m_1+1\big\}. \end{aligned}$$

Let $\mathcal{T}'$ be a one-way transducer equivalent to $\mathcal{T}$, and consider a successful run $\lambda^{(m_1,m_2)}$ of $\mathcal{T}'$ on the input $u^{(m_1,m_2)}$. Since $\mathcal{T}'$ has finitely many states, we can find a large enough number m and two positions $x'_1 < x'_2$ both in $X_1^{(m,m)}$, such that $L'_1 = [x'_1,x'_2]$ is a loop of $\lambda^{(m,m)}$. Similarly, we can find two positions $x'_3 < x'_4$ both in $X_2^{(m,m)}$, such that $L'_2 = [x'_3,x'_4]$ is a loop of $\lambda^{(m,m)}$. Clearly, L'_1 and L'_2 are also loops of $\rho^{(m,m)}$: indeed, L'_1 (resp. L'_2) consists of $k_1 \leqslant m$ (resp. $k_2 \leqslant m$) copies of L_1 (resp. L_2) in $\rho^{(m,m)}$. In particular, for all $m_1, m_2 \in \mathbb{N}$ we have:

$$\begin{aligned} \mathsf{pump}^{m_1+1}_{L'_1}(\mathsf{pump}^{m_2+1}_{L'_2}(u^{(m,m)})) &= u^{(f(m_1),g(m_2))} \\ \mathsf{pump}^{m_1+1}_{L'_1}(\mathsf{pump}^{m_2+1}_{L'_2}(\rho^{(m,m)})) &= \rho^{(f(m_1),g(m_2))} \\ \mathsf{pump}^{m_1+1}_{L'_1}(\mathsf{pump}^{m_2+1}_{L'_2}(\lambda^{(m,m)})) &= \lambda^{(f(m_1),g(m_2))}. \end{aligned}$$

where $f(m_1) = k_1 \cdot m_1 + m$, $g(m_2) = k_2 \cdot m_2 + m$.

Now we observe that the run $\lambda^{(f(m_1),g(m_2))}$ of $\mathcal{T}'$ produces the same output as the run $\rho^{(f(m_1),g(m_2))}$ of $\mathcal{T}$ — this holds thanks to the fact that the transducers are functional, otherwise it may happen that the pumped runs $\lambda^{(f(m_1),g(m_2))}$ and $\rho^{(f(m_1),g(m_2))}$ produce different outputs. Let us denote this output by $w^{(f(m_1),g(m_2))}$. Below, we show two possible factorizations of $w^{(f(m_1),g(m_2))}$ based on the shapes of the pumped runs $\lambda^{(f(m_1),g(m_2))}$ and $\rho^{(f(m_1),g(m_2))}$. For the first factorization, we recall that L'_2 precedes L'_1, according to the ordering of positions, and that the run $\lambda^{(f(m_1),g(m_2))}$ is left-to-right. We thus obtain

$$w^{(f(m_1),g(m_2))} \ =\ w_0\ \boldsymbol{w_1^{m_2}}\ w_2\ \boldsymbol{w_3^{m_1}}\ w_4 \tag{1}$$

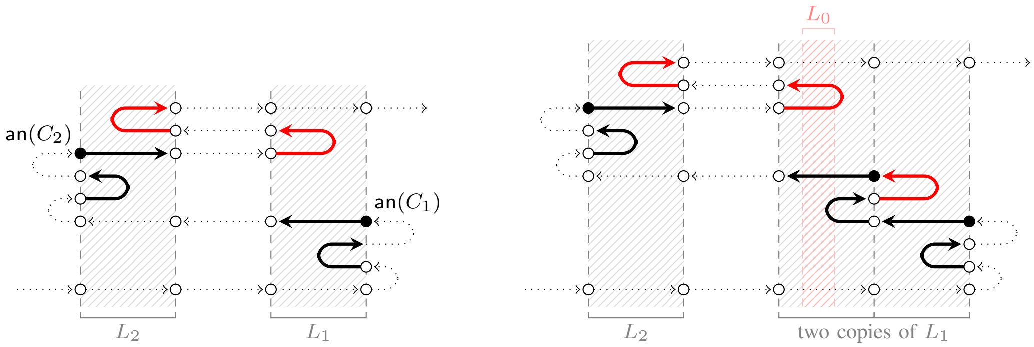


Fig. 11. An inversion (L_1, C_1, L_2, C_2) whose pairs (L_i, C_i) are not output-minimal. The red parts produce long outputs, and lie outside $\rho[\mathsf{an}(C_1), \mathsf{an}(C_2)]$.

where

- w_0 is the output produced by the prefix of $\lambda^{(m,m)}$ up to the left border of L'_2,
- w_1 is the output produced by the (unique) factor of $\lambda^{(m,m)}$ intercepted by L'_2,
- w_2 is the output produced by the factor of $\lambda^{(m,m)}$ between the right border of L'_2 and the left border of L'_1,
- w_3 is the output produced by the (unique) factor of $\lambda^{(m,m)}$ intercepted by L'_1,
- w_4 is the output produced by the suffix of $\lambda^{(m,m)}$ after the right border of L'_1.

For the second factorization, we consider L'_1 and L'_2 as loops of $\rho^{(m,m)}$. We denote by ℓ'_1 (resp. ℓ'_2) the anchor of the component C_1 (resp. C_2) of L'_1 (resp. L'_2). By assumption we have $\ell'_1 \trianglelefteq \ell'_2$. Applying Proposition 11 we get:

$$w^{(f(m_1),g(m_2))} \;=\; v_0^{(m_1,m_2)}\ \boldsymbol{v_1^{m_1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2}}\ v_4^{(m_1,m_2)} \tag{2}$$

where

- $v_0^{(m_1,m_2)}$ is the output produced by the prefix of $\rho^{(m,m)}$ that ends at ℓ'_1 (note that this word may depend on the parameters m_1, m_2, since the loops L'_1 and L'_2 may be traversed several times before reaching the location ℓ'_1),
- $v_1 = \mathsf{out}(\mathsf{tr}(C_1))$ (this word does not depend on m_1, m_2),
- $v_2^{(m_1,m_2)}$ is the output produced by the factor of $\rho^{(m,m)}$ between ℓ'_1 and ℓ'_2,
- $v_3 = \mathsf{out}(\mathsf{tr}(C_2))$,
- $v_4^{(m_1,m_2)}$ is the output produced by the suffix of $\rho^{(m,m)}$ that starts at ℓ'_2.

Putting together Eqs. (1) and (2), we get

$$v_0^{(m_1,m_2)}\ \boldsymbol{v_1^{m_1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2}}\ v_4^{(m_1,m_2)} \quad=\quad w_0\ \boldsymbol{w_1^{m_2}}\ w_2\ \boldsymbol{w_3^{m_1}}\ w_4. \tag{3}$$

We recall that the words v_1, v_3 are non-empty, since they are outputs of traces of components that form an inversion. This allows us to apply Lemma 19, which shows that the word $\boldsymbol{v_1}\ \boldsymbol{v_1^{m_1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2}}\ \boldsymbol{v_3}$ has period $\gcd(|v_1|, |v_3|)$, for all $m_1, m_2 \in \mathbb{N}$. Note that the latter period still depends on $\mathcal{T}'$, since the words v_1 and v_3 were constructed from the loops L'_1 and L'_2, that are both loops of the run $\lambda^{(m,m)}$ of $\mathcal{T}'$. However, Proposition 11 tells us that the word v_1 (resp. v_3) is an iteration of the output $\mathsf{out}(\mathsf{tr}(C_1))$ of the component C_1 of L_1 (resp. the output $\mathsf{out}(\mathsf{tr}(C_2))$ of the component C_2 of L_2). By Lemma 17, this implies that the period of $\boldsymbol{v_1}\ \boldsymbol{v_1^{m_1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2}}\ \boldsymbol{v_3}$ divides both $|\mathsf{out}(\mathsf{tr}(C_1))|$ and $|\mathsf{out}(\mathsf{tr}(C_2))|$.

In a similar way, we recall from Proposition 11 that all the words $\boldsymbol{v_1}\ \boldsymbol{v_1^{m_1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2}}\ \boldsymbol{v_3}$ are obtained by iterating suitable factors inside $\mathsf{out}(\mathsf{tr}(C_1))\ \mathsf{out}(\rho[\mathsf{an}(C_1), \mathsf{an}(C_2)])\ \mathsf{out}(\mathsf{tr}(C_2))$: more precisely, by iterating n_1 (resp. n_2) times the output traces of the components of L_1 (resp. of L_2), where $n_1 = f(m_1)$ (resp. $n_2 = g(m_2)$). Since the periodicity property holds for infinitely many n_1 and, independently, for infinitely many n_2, we know from Theorem 18 that it also holds for all $n_1, n_2 \in \mathbb{N}$, and in particular, for $n_1 = n_2 = 0$. This allows us to conclude that the word

$$\mathsf{out}\big(\mathsf{tr}(C_1)\big)\ \mathsf{out}\big(\rho[\mathsf{an}(C_1), \mathsf{an}(C_2)]\big)\ \mathsf{out}\big(\mathsf{tr}(C_2)\big)$$

is periodic with period p that divides both $|\mathsf{out}(\mathsf{tr}(C_1))|$ and $|\mathsf{out}(\mathsf{tr}(C_2))|$.

It remains to prove the second claim of the proposition, which bounds the period by the constant $\boldsymbol{B}$. This requires a refinement of the previous arguments that involves pumping the run ρ simultaneously on three different loops.

Recall that the period p for the word $\mathsf{out}\big(\mathsf{tr}(C_1)\big)\ \mathsf{out}\big(\rho[\mathsf{an}(C_1), \mathsf{an}(C_2)]\big)\ \mathsf{out}\big(\mathsf{tr}(C_2)\big)$ was obtained by considering a run $\rho^{(m_1,m_2)}$ where the loops L_1 and L_2 have been pumped m_1 and m_2 times, respectively. To bound the period, we need to consider inversions that are formed by output-minimal pairs. As already explained, we cannot assume that the inversion (L_1, C_1, L_2, C_2) contains an output-minimal pair. For example, the left part of Fig. 11 represents a situation where both pairs

(L_1, C_1) and (L_2, C_2) of the inversion are not output-minimal. Nonetheless, in the pumped run $\rho^{(2,1)}$ we do find inversions with output-minimal pairs. For example, as suggested by the right part of Fig. 11, we can consider the leftmost and rightmost occurrences of L_1 in $\rho^{(2,1)}$, denoted as $\overleftarrow{L}_1$ and $\overrightarrow{L}_1$, respectively. Let (L_0, C_0) be any *output-minimal* pair such that L_0 is an idempotent loop, $\mathsf{out}(\mathsf{tr}(C_0)) \neq \varepsilon$, and either $(L_0, C_0) = (\overleftarrow{L}_1, C_1)$ or $(L_0, C_0) \sqsubset (\overleftarrow{L}_1, C_1)$ — such a loop L_0 is suggestively represented in the figure by the red vertical stripe.

We claim that either (L_0, C_0, L_2, C_2) or $(\overrightarrow{L}_1, C_1, L_0, C_0)$ is an inversion of the run $\rho^{(2,1)}$, depending on whether the anchor of C_0 inside L_0 occurs before or after the anchor of C_2 inside L_2. First, note that all the loops L_0, L_2, $\overrightarrow{L}_1$ are idempotent and non-overlapping; more precisely, we have $\max(L_2) \leqslant \min(L_0)$ and $\max(L_0) \leqslant \min(\overrightarrow{L}_1)$. Moreover, the trace outputs for the pairs (L_0, C_0), (L_2, C_2), $(\overrightarrow{L}_1, C_1)$ are non-empty. So it remains to distinguish the two cases based on the ordering of the anchors of C_0, C_1, C_2 inside the loops L_0, $\overrightarrow{L}_1$, L_2, respectively. We denote those anchors by ℓ_0, ℓ_1, ℓ_2. If $\ell_0 \trianglelefteq \ell_2$, then (L_0, C_0, L_2, C_2) is clearly an inversion. Otherwise, because $(\overrightarrow{L}_1, C_1, L_2, C_2)$ is an inversion, we know that $\ell_1 \trianglelefteq \ell_2 \trianglelefteq \ell_0$, and hence $(\overrightarrow{L}_1, C_1, L_0, C_0)$ is an inversion.

Now, we know that $\rho^{(2,1)}$ contains the inversion $(\overrightarrow{L}_1, C_1, L_2, C_2)$, but also an inversion with an output-minimal pair (L_0, C_0), where L_0 is strictly between $\overrightarrow{L}_1$ and L_2. For all m_0, m_1, m_2, we define $\rho^{(m_0,m_1,m_2)}$ as the run obtained from $\rho^{(2,1)}$ by pumping m_0, m_1, m_2 times the loops $L_0, \overrightarrow{L}_1, L_2$, respectively. Since the output of the run $\rho^{(m_0,m_1,m_2)}$ contains many repetitions of the trace output $\mathsf{out}(\mathsf{tr}(C_0))$ of C_0 inside L_0, and since these repetitions occur as factors of the output produced inside the inversion $(\overrightarrow{L}_1, C_1, L_2, C_2)$, their period p' divides $|\mathsf{out}(\mathsf{tr}(C_0))|$, $|\mathsf{out}(\mathsf{tr}(C_1))|$, and $|\mathsf{out}(\mathsf{tr}(C_2))|$ (due to Lemma 17). By Theorem 18, we deduce that the word $\mathsf{out}\big(\mathsf{tr}(C_1)\ \rho[\mathsf{an}(C_1), \mathsf{an}(C_2)]\ \mathsf{tr}(C_2)\big)$ has period p' as well. To conclude the proof, it suffices to recall Lemma 21, saying that the length of $\mathsf{out}(\mathsf{tr}(C_0))$, and hence the period p', is bounded by $\boldsymbol{B}$. □

Lemma 19. *Consider a word equation of the form*

$$v_0^{(m_1,m_2)}\ \boldsymbol{v_1^{m_1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2}}\ v_4^{(m_1,m_2)} \ = \ w_0\ \boldsymbol{w_1^{m_2}}\ w_2\ \boldsymbol{w_3^{m_1}}\ w_4$$

where m_1, m_2 are the unknowns, v_1, v_3 are non-empty words, and $v_0^{(m_1,m_2)}, v_2^{(m_1,m_2)}, v_4^{(m_1,m_2)}$ are words that may contain some factors of the form v^{m_1} or v^{m_2}, for some v. If the above equation holds for all $m_1, m_2 \in \mathbb{N}$, then the words $\boldsymbol{v_1}\ \boldsymbol{v_1^{m_1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2}}\ \boldsymbol{v_3}$ are periodic with period $\gcd(|v_1|, |v_3|)$, for all $m_1, m_2 \in \mathbb{N}$.

Proof. The idea of the proof is to let the parameters m_1, m_2 of the equation grow independently, and exploit Fine and Wilf's theorem (Lemma 17) a certain number of times to establish periodicities in overlapping factors of the considered words.

We begin by fixing m_1 large enough so that the factor $\boldsymbol{v_1^{m_1}}$ of the left hand-side of the equation is longer than $|w_0| + |w_1|$ (this is possible because v_1 is non-empty). Now, if we let m_2 grow arbitrarily large, we see that the length of the periodic word $\boldsymbol{w_1^{m_2}}$ is almost equal to the length of the left hand-side term $v_0^{(m_1,m_2)}\ \boldsymbol{v_1^{m_1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2}}\ v_4^{(m_1,m_2)}$: indeed, the difference in length is given by the constant $|w_0| + |w_2| + m_1 \cdot |w_3| + |w_4|$. In particular, this implies that $\boldsymbol{w_1^{m_2}}$ covers arbitrarily long prefixes of $\boldsymbol{v_1}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2+1}}$, which in its turn contains long repetitions of the word v_3. Hence, by Lemma 17, the word $\boldsymbol{v_1}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2+1}}$ has period $|v_3|$.

We remark that the periodicity shown so far holds for infinitely many m_1 and for all but finitely many m_2, where the threshold for m_2 depends on m_1: once m_1 is fixed, m_2 needs to be larger than $f(m_1)$, for a suitable function f. In fact, using Theorem 18, we can show that the periodicity holds even when m_2 ranges over all natural numbers. To see this, we introduce the following shorthand: given a word w and a rational number $r = \frac{n}{|w|}$, with $n \in \mathbb{N}$, we denote by w^r the word $w^{\lfloor r \rfloor}\ w'$, where w' is the prefix of w of length $|w| \cdot (r - \lfloor r \rfloor)$. We then state the periodicity property for $\boldsymbol{v_1}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2+1}}$ as an equation of the form

$$\boldsymbol{v_1}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2+1}} \ = \ \boldsymbol{v}^{\frac{g(m_2)}{|v|}}$$

which, once m_1 is fixed, must hold for all but finitely many m_2, for a suitable word v of the same length as v_3, and for a suitable linear function $g : \mathbb{N} \to \mathbb{N}$. More precisely, $g(m_2)$ gives the length of the left hand-side of the equation. The above equation can be easily rewritten so as to highlight all the repetitions that depend on m_2, including those that are hidden inside the term $v_2^{(m_1,m_2)}$. Note that we cannot apply Theorem 18 yet, since the repetitions in the right hand-side of the equation may be fractional. If this is the case, however, it means that the left hand-side of the equation contains a repetition of the form w^{m_2}, for some word w whose length is not multiple of $|v|$. By Fine and Wilf's theorem (Lemma 17), we know that the period of the left hand-side is in fact smaller, i.e. $\gcd(|v_3|, |w|)$. We can then replace the right hand-side of the equation with an exact repetition of a word v' shorter than v. This enables the application of Theorem 18, which implies that the equation holds for all $m_2 \in \mathbb{N}$. In this way we have shown that the word $\boldsymbol{v_1}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3^{m_2+1}}$ has period $|v_3|$ for all $m_2 \in \mathbb{N}$.

We could also apply a symmetric reasoning, by fixing m_2 and by letting m_1 grow arbitrarily large. Doing so, we prove that for a large enough m_2 and for all but finitely many m_1, the word $\boldsymbol{v_1^{m_1+1}}\ v_2^{(m_1,m_2)}\ \boldsymbol{v_3}$ is periodic with period $|v_1|$. As before, this can be strengthened to hold for all $m_1 \in \mathbb{N}$, independently of the choice of m_2.

Putting together the results proven so far, we get that for all but finitely many m_1, m_2,

$$\underbrace{\boldsymbol{v_1^{m_1}} \cdot \overbrace{\boldsymbol{v_1} \;\cdot\; v_2^{(m_1,m_2)}}^{\text{period } |v_3|}}_{\text{period } |v_1|}\!\!\overbrace{\;\cdot\; \boldsymbol{v_3}}\cdot \boldsymbol{v_3^{m_2}}\,.$$

Finally, we observe that the prefix $\boldsymbol{v_1^{m_1+1}} \cdot v_2^{(m_1,m_2)} \cdot \boldsymbol{v_3}$ and the suffix $\boldsymbol{v_1} \cdot v_2^{(m_1,m_2)} \cdot \boldsymbol{v_3^{m_2+1}}$ share a common factor of length at least $|v_1| + |v_3|$. By Lemma 17, we derive that $\boldsymbol{v_1^{m_1+1}} \cdot v_2^{(m_1,m_2)} \cdot \boldsymbol{v_3^{m_2+1}}$ has period $\gcd(|v_1|, |v_3|)$. Finally, by exploiting again Theorem 18, we generalize this periodicity property to all $m_1, m_2 \in \mathbb{N}$. ☐

Lemma 21. *For every output-minimal pair (L, C), $|\mathsf{out}(\mathsf{tr}(C))| \leqslant \boldsymbol{B}$.*

Proof. Consider a pair (L, C) consisting of an idempotent loop $L = [x_1, x_2]$ and a component C of L. We suppose that the length of $\mathsf{out}(\mathsf{tr}(C))$ exceeds $\boldsymbol{B}$ and we claim that (L, C) is not output-minimal.

Recall that $\mathsf{tr}(C)$ is a concatenation of (L, C)-factors, say, $\mathsf{tr}(C) = \beta_1 \cdots \beta_k$. Let ℓ_1 (resp. ℓ_2) be the first (resp. last) location that is visited by these factors. Further let $K = [\ell_1, \ell_2]$ and $Z = K \cap (L \times \mathbb{N})$. By construction, the subrun $\rho \mid Z$ can be seen as a concatenation of the factors $\beta_1, \ldots, \beta_k$, possibly in a different order than that of $\mathsf{tr}(C)$. This implies that $|\mathsf{out}(\rho \mid Z)| > \boldsymbol{B}$.

By Theorem 13, we know that there exist an idempotent loop $L' \subsetneq L$ and a component C' of L' such that $\mathsf{an}(C') \in K$ and $\mathsf{out}(\mathsf{tr}(C')) \neq \varepsilon$. In particular, the (L', C')-factor that starts at the location $\mathsf{an}(C')$ is entirely contained in some (L, C)-factor. This implies that $(L', C') \sqsubset (L, C)$, and thus (L, C) is not output-minimal. ☐

Proposition 26. *If ρ satisfies the periodicity property stated in* **P2** *and $\ell \trianglelefteq \ell'$ are two locations in the same S^*-class, then $\mathsf{out}\big(\rho[\ell, \ell']\big)$ has period at most $\boldsymbol{B}$.*

Proof. The claim for $\ell = \ell'$ holds trivially, so we assume that $\ell \vartriangleleft \ell'$. We know that ℓ, ℓ' belong to the same non-singleton S^*-class. By definition of S, the run ρ contains some inversions (L_0, C_0, L_1, C_1), (L_2, C_2, L_3, C_3), $\ldots$, $(L_{2k}, C_{2k}, L_{2k+1}, C_{2k+1})$ such that $\mathsf{an}(C_0) \trianglelefteq \ell \vartriangleleft \ell' \trianglelefteq \mathsf{an}(C_{2k+1})$ and $\mathsf{an}(C_{2i}) \trianglelefteq \mathsf{an}(C_{2i+2}) \trianglelefteq \mathsf{an}(C_{2i+1}) \trianglelefteq \mathsf{an}(C_{2i+3})$ for all $i = 0, \ldots, k-1$. Without loss of generality we can assume that every inversion $(L_{2i}, C_{2i}, L_{2i+1}, C_{2i+1})$ is *maximal* in the following sense: there is no other inversion $(L, C, L', C') \neq (L_{2i}, C_{2i}, L_{2i+1}, C_{2i+1})$ such that $\mathsf{an}(C) \trianglelefteq \mathsf{an}(C_{2i}) \trianglelefteq \mathsf{an}(C_{2i+1}) \trianglelefteq \mathsf{an}(C')$.

We introduce the following shorthands for all $i = 0, \ldots, 2k+1$: $\ell_i = \mathsf{an}(C_i)$, $v_i = \mathsf{out}(\mathsf{tr}(C_i))$, and $p_i = |v_i|$. By Property P2, we know that $v_{2i}\ \mathsf{out}(\rho[\ell_{2i}, \ell_{2i+1}])\ v_{2i+1}$ has period at most $\boldsymbol{B}$ that divides both p_{2i} and p_{2i+1}.

In order to show that $\mathsf{out}\big(\rho[\ell, \ell']\big)$ has period at most B, it suffices to prove the following claim by induction on i:

Claim. *For all $i = 0, \ldots, k$, the period of $\mathsf{out}\big(\rho[\ell_0, \ell_{2i+1}]\big)\ v_{2i+1}$ divides p_{2i+1} and is bounded by $\boldsymbol{B}$.*

Proof of claim. The base case $i = 0$ follows from Property P2, since (L_0, C_0, L_1, C_1) is an inversion. For the inductive step, we assume that the claim holds for $i < k$ and we prove it for $i+1$. We factorize our word as follows:

$$\mathsf{out}\big(\rho[\ell_0, \ell_{2i+3}]\big)\ v_{2i+3} \;=\; \underbrace{\mathsf{out}\big(\rho[\ell_0, \ell_{2i+2}]\big)\ \overbrace{\mathsf{out}\big(\rho[\ell_{2i+2}, \ell_{2i+1}]\big)}^{\text{periods } p_{2i+2} \text{ and } p_{2i+3}}}_{\text{period } p_{2i+1}}\!\overbrace{\mathsf{out}\big(\rho[\ell_{2i+1}, \ell_{2i+3}]\big)\ v_{2i+3}}\,.$$

By the inductive hypothesis, the output produced between ℓ_0 and ℓ_{2i+1}, even extended to the right with the trace output v_{2i+1}, has period that divides p_{2i+1}. Moreover, because $(L_{2i+2}, C_{2i+2}, L_{2i+3}, C_{2i+3})$ is an inversion, the output produced between the locations $\ell_{2i+2} = \mathsf{an}(C_{2i+2})$ and $\ell_{2i+3} = \mathsf{an}(C_{2i+3})$, extended to the left with v_{2i+2} and to the right with v_{2i+3}, has period that divides both p_{2i+2} and p_{2i+3}. This does not suffice yet to apply Fine-Wilf's theorem so as to derive a suitable period of $\mathsf{out}\big(\rho[\ell_0, \ell_{2i+3}]\big)\ v_{2i+3}$, since the common factor $\mathsf{out}\big(\rho[\ell_{2i+2}, \ell_{2i+1}]\big)$ might be too short. The key argument here is that the interval $[\ell_{2i+2}, \ell_{2i+1}]$ is covered by the inversion $(L_{2i+2}, C_{2i+2}, L_{2i+1}, C_{2i+1})$, which is different from the previous ones.

For this, we have to prove that the anchors $\mathsf{an}(C_{2i+2})$ and $\mathsf{an}(C_{2i+1})$ are correctly ordered w.r.t. $\trianglelefteq$ and the ordering of positions (recall Definition 14). First, we have $\mathsf{an}(C_{2i+2}) \trianglelefteq \mathsf{an}(C_{2i+1})$ by assumption. Now we prove that $\mathsf{an}(C_{2i+1})$ is strictly to the left of $\mathsf{an}(C_{2i+2})$, according to the ordering of positions. By way of contradiction, suppose that this is not the case, namely, $\mathsf{an}(C_{2i+1}) = (x_{2i+1}, y_{2i+1})$, $\mathsf{an}(C_{2i+2}) = (x_{2i+2}, y_{2i+2})$, and $x_{2i+1} > x_{2i+2}$. Because $(L_{2i}, C_{2i}, L_{2i+1}, C_{2i+1})$ and $(L_{2i+2}, C_{2i+2}, L_{2i+3}, C_{2i+3})$ are inversions, we know that $\mathsf{an}(C_{2i+3})$ is to the left of $\mathsf{an}(C_{2i+2})$ and $\mathsf{an}(C_{2i+1})$ is to the left of $\mathsf{an}(C_{2i})$. This implies that $\mathsf{an}(C_{2i+3})$ is to the left of $\mathsf{an}(C_{2i})$, and hence $(L_{2i}, C_{2i}, L_{2i+3}, C_{2i+3})$ is also an inversion. But this would contradict the maximality of $(L_{2i}, C_{2i}, L_{2i+1}, C_{2i+1})$, which was assumed at the beginning of the proof. ☐

Now that we know that $\mathsf{an}(C_{2i+2})$ and $\mathsf{an}(C_{2i+1})$ are correctly ordered w.r.t. $\trianglelefteq$ and the ordering of positions, we recall that the trace outputs v_{2i+1} and v_{2i+2} are non-empty. This implies that $(L_{2i+2}, C_{2i+2}, L_{2i+1}, C_{2i+1})$ is an inversion. Moreover, the latter inversion covers the interval of locations $[\ell_{2i+2}, \ell_{2i+1}]$. By Property P2, the word v_{2i+2} $\mathsf{out}(\rho[\ell_{2i+2}, \ell_{2i+1}])$ v_{2i+1} has period that divides both p_{2i+2} and p_{2i+1}.

Summing up, we have:

1) $w_1 = \mathsf{out}\big(\rho[\ell_0, \ell_{2i+1}]\big)\ v_{2i+1}$ has period p_{2i+1},
2) $w_2 = v_{2i+2}\ \mathsf{out}\big(\rho[\ell_{2i+2}, \ell_{2i+1}]\big)\ v_{2i+1}$ has period $p = \gcd(p_{2i+2}, p_{2i+1})$,
3) $w_3 = v_{2i+2}\ \mathsf{out}\big(\rho[\ell_{2i+2}, \ell_{2i+3}]\big)\ v_{2i+3}$ has period $p' = \gcd(p_{2i+2}, p_{2i+3})$.

We are now ready to exploit our slightly stronger variant of Fine-Wilf's theorem, that is, Lemma 17.

Let $w = \mathsf{out}\big(\rho[\ell_{2i+2}, \ell_{2i+1}]\big)\ v_{2i+1}$ be the common suffix of w_1 and w_2. From 1. and 2., we know that the latter words have period p_{2i+1} and $p = \gcd(p_{2i+2}, p_{2i+1})$, respectively. Moreover, since p divides $|w_2| - |w|$ ($= |v_{2i+2}|$), w is also a prefix of w_2. For the same reason, we also know that $|w| \geqslant |v_{2i+1}| = p_{2i+1} = p_{2i+1} + p - \gcd(p_{2i+1}, p)$ (the latter equality follows from the fact that p divides p_{2i+1}). Thus, by applying Lemma 17 to $w_1 = w_1'\, w$ and $w_2 = w\, w_2''$, using w as common factor, we obtain that

4) $w_4 = w_1'\, w\, w_2'' = \mathsf{out}\big(\rho[\ell_0, \ell_{2i+2}]\big)\ v_{2i+2}\ \mathsf{out}\big(\rho[\ell_{2i+2}, \ell_{2i+1}]\big)\ v_{2i+1}$ has period p.

Now, from 2. and 3., we know that the words w_2 and w_3 have periods p and p', respectively, and contain v_{2i+2} as factor. Moreover, the length of the factor v_{2i+2} is a multiple of both periods p and p', and hence $|v_{2i+2}| \geqslant p + p' - \gcd(p, p')$ (this is folklore, and follows from basic facts in number theory, such as $q \cdot q' \geqslant q + q' - 1$ for all $q, q' \in \mathbb{N}$). From Lemma 17 we derive that

5) $w_5 = v_{2i+2}\ \mathsf{out}\big(\rho[\ell_{2i+2}, \ell_{2i+3}]\big)\ v_{2i+3}$ has period $p'' = \gcd(p_{2i+1}, p_{2i+2}, p_{2i+3})$.

In a similar way, from 4. and 5., using again v_{2i+2} as common factor of w_4 and w_5, we derive

6) $w_6 = \mathsf{out}\big(\rho[\ell_0, \ell_{2i+2}]\big)\ v_{2i+2}\ \mathsf{out}\big(\rho[\ell_{2i+2}, \ell_{2i+3}]\big)\ v_{2i+3}$ has period p''.

Finally, the periodicity is not affected when we remove factors of length multiple than the period. In particular, by removing the factor v_{2i+2} from w_6, we obtain the word $\mathsf{out}\big(\rho[\ell_0, \ell_{2i+3}]\big)\ v_{2i+3}$, whose period still divides p_{2i+3}. This proves the claim for the inductive step, and completes the proof of the proposition. □

Lemma 28. *If $K = [\ell, \ell']$ is a non-singleton S^*-class, then $\rho[\ell_1, \ell_2]$ is a block, where $[\ell_1, \ell_2] = \mathsf{block}(K)$.*

Proof. Let $K = [\ell, \ell']$ and $\mathsf{block}(K) = [\ell_1, \ell_2]$, with $\ell_i = (x_i, y_i)$ for both $i = 1, 2$, and let $\mathsf{an}(K)$ and $X_{\mathsf{an}(K)}$ be the sets given in Definition 27.

We begin by observing that the factor $\rho[\ell_1, \ell]$ between the first location of the block and the first location of the equivalence class lies entirely to the right of position x_1. Indeed, if this were not the case, there would exist another location $\ell_1' = (x_1, y_1{+}1)$, on the same position x_1 as ℓ_1 but at a higher level, such that $\ell_1 \lhd \ell_1' \unlhd \ell$. But this would contradict Definition 27. In a similar way one verifies that the factor $\rho[\ell', \ell_2]$ lies to the left of x_2.

Next, we prove that the output produced by the factor $\rho[\ell_1, \ell_2]$ is quasi-periodic. By Definition 27, we have $\ell_1 \unlhd \ell \lhd \ell' \unlhd \ell_2$, and by Proposition 26 we know that $\mathsf{out}(\rho[\ell, \ell'])$ is periodic with period at most $\boldsymbol{B}$. So it suffices to bound the length of the words $\mathsf{out}(\rho[\ell_1, \ell])$ and $\mathsf{out}(\rho[\ell', \ell_2])$. We shall focus on the former word, as the arguments for the latter are similar. As usual, the idea is to apply a Ramsey-type argument.

Suppose, by way of contradiction, that the length of $|\mathsf{out}(\rho[\ell_1, \ell])| > \boldsymbol{B}$. We head towards finding a location $\ell'' \lhd \ell$ that is S^*-equivalent to ℓ, thus contradicting the fact that ℓ is the first location of the equivalence class K. Recall that the factor $\rho[\ell_1, \ell]$ lies entirely to the right of the position x_1 of ℓ_1, so $|\mathsf{out}(\rho[\ell_1, \ell])| > \boldsymbol{B}$ is equivalent to saying $|\mathsf{out}(\rho \mid Z)| > \boldsymbol{B}$, where $Z = [\ell_1, \ell] \cap \big([x_1, \omega] \times \mathbb{N}\big)$. Theorem 13 implies the existence of an idempotent loop L and a component C such that

- $\min(L) > x_1$,
- $\ell_1 \lhd \mathsf{an}(C) \lhd \ell$,
- $\mathsf{out}(\mathsf{tr}(C)) \neq \varepsilon$.

Let $\ell'' = \mathsf{an}(C)$. By construction, x_1 is the leftmost position of all the locations of the class $K = [\ell, \ell']$ that are also anchors of components of inversions. Thus there exist an inversion (L_1, C_1, L_2, C_2) and a location $\ell''' = (x_1, y''') \in K$ such that $\ell''' = \mathsf{an}(C_i)$ for some $i \in \{1, 2\}$. Since $\ell'' \lhd \ell \unlhd \ell'''$ and the position of ℓ'' is to the right of x_1, we know that (L, C, L_i, C_i) is also an inversion, and hence $\ell''\ \mathsf{S}^*\ \ell'''\ \mathsf{S}^*\ \ell$. But since $\ell'' \neq \ell$, we get a contradiction with the assumption that ℓ is the first location of a S^*-class. In this way we have shown that $|\mathsf{out}(\rho[\ell_1, \ell])| \leqslant \boldsymbol{B}$.

It remains to bound the lengths of the outputs produced by the subruns $\rho \mid Z^{\leftarrow}$ and $\rho \mid Z^{\rightarrow}$, where $Z^{\leftarrow} = [\ell_1, \ell_2] \cap \big([0, x_1] \times \mathbb{N}\big)$ and $Z^{\rightarrow} = [\ell_1, \ell_2] \cap \big([x_2, \omega] \times \mathbb{N}\big)$. As usual, we consider only one of the two symmetric cases. Suppose, by way of contradiction, that $|\mathsf{out}(\rho \mid Z^{\leftarrow})| > \boldsymbol{B}$. By Theorem 13, there exist an idempotent loop L and a component C of L such that

- $\max(L) < x_1$,
- $\ell_1 \lhd \mathsf{an}(C) \lhd \ell_2$,
- $\mathsf{out}(\mathsf{tr}(C)) \neq \varepsilon$.

Let $\ell'' = \mathsf{an}(C)$. By following the same line of reasoning as before, we recall that ℓ is the first location of the non-singleton class K. From this we derive the existence an inversion (L_1, C_1, L_2, C_2) such that $\ell = \mathsf{an}(C_1)$. We claim that $\ell \trianglelefteq \ell''$. Indeed, if this were not the case, then, because ℓ'' is strictly to the left of x_1 and ℓ is to the right of x_1, there would exist a location ℓ'_1 between ℓ'' and ℓ that lies at position x_1. But $\ell_1 \lhd \ell'' \trianglelefteq \ell'_1 \trianglelefteq \ell$ would contradict the fact that ℓ_1 is the *latest* location before ℓ that lies at the position $x_1 = \min(X_{\mathsf{an}(K)})$. Now that we know that $\ell \trianglelefteq \ell''$ and that ℓ'' is to the left of x_1, we observe that (L_1, C_1, L, C) is also an inversion, and hence $\ell'' \in \mathsf{an}(K)$. Since ℓ'' is strictly to the left of x_1, we get a contradiction with the definition of x_1 as leftmost position of the locations of K that are anchors of components of inversions. We must conclude that $|\mathsf{out}(\rho \mid Z^{\leftarrow})| \leqslant \boldsymbol{B}$.

This completes the proof that $\rho \mid \mathsf{block}(K)$ is a block. $\square$

Lemma 29. *Suppose that K_1 and K_2 are two different non-singleton S^*-classes such that $\ell \lhd \ell'$ for all $\ell \in K_1$ and $\ell' \in K_2$. Let $\mathsf{block}(K_1) = [\ell_1, \ell_2]$ and $\mathsf{block}(K_2) = [\ell_3, \ell_4]$, with $\ell_2 = (x_2, y_2)$ and $\ell_3 = (x_3, y_3)$. Then $x_2 < x_3$.*

Proof. Suppose by contradiction that K_1 and K_2 are as in the statement, but $x_2 \geqslant x_3$. By Definition 27, $x_2 = \max(X_{\mathsf{an}(K_1)})$ and $x_3 = \min(X_{\mathsf{an}(K_2)})$. This implies the existence of some inversions (L_1, C_1, L_2, C_2) and (L_3, C_3, L_4, C_4) such that $\mathsf{an}(C_i) = (x_2, y)$ for some $i \in \{1, 2\}$ and $\mathsf{an}(C_j) = (x_3, y')$ for some $j \in \{3, 4\}$. Moreover, since $\mathsf{an}(C_i) \trianglelefteq \mathsf{an}(C_j)$ and $x_2 \geqslant x_3$, we know that (L_i, C_i, L_j, C_j) is also an inversion. But this means that $K_1 = K_2$. $\square$

Lemma 30. *Let $\rho[\ell_1, \ell_2]$ be a factor of ρ that does not overlap any S^*-block, with $\ell_1 = (x_1, y_1)$, $\ell_2 = (x_2, y_2)$, and $x_1 < x_2$. Then $\rho[\ell_1, \ell_2]$ is a diagonal.*

Proof. Suppose by contradiction that there is some $x \in [x_1, x_2]$ such that, for all locations $\ell = (x, y)$ between ℓ_1 and ℓ_2, one of the following conditions holds:

1) $|\mathsf{out}(\rho \mid Z_\ell^{\nwarrow})| > \boldsymbol{B}$, where $Z_\ell^{\nwarrow} = [\ell, \ell_2] \cap \big([0, x] \times \mathbb{N}\big)$,
2) $|\mathsf{out}(\rho \mid Z_\ell^{\searrow})| > \boldsymbol{B}$, where $Z_\ell^{\searrow} = [\ell_1, \ell] \cap \big([x, \omega] \times \mathbb{N}\big)$.

We claim first that for each condition above there is some level y at which it holds. Observe that for the highest location ℓ of the run at position x, the set $Z_\ell^{\nwarrow}$ is empty, since the outgoing transition at ℓ is rightward. So condition 1 is trivially violated at ℓ as above, hence condition 2 holds by the initial assumption. Symmetrically, condition 1 holds at the lowest location of the run at position x. Let us now compare, for each condition, the levels where it holds.

Clearly, the lower the level of the location ℓ, the easier it is to satisfy condition 1, and symmetrically for condition 2. So, let $\ell = (x, y)$ (resp. $\ell' = (x, y')$) be the highest (resp. lowest) location at position x that satisfies condition 1 (resp. condition 2).

We claim that $y \geqslant y'$. For this, we first observe that $y \geqslant y' - 1$, since otherwise there would exist a location $\ell = (x, y'')$, with $y < y'' < y'$, violating both conditions 1 and 2. Moreover, y must be odd, otherwise the transition departing from $\ell = (x, y)$ would be rightward oriented and the location $\ell'' = (x, y + 1)$ would still satisfy condition 1, contradicting the fact that $\ell = (x, y)$ was chosen to be the highest location. For similar reasons, y' must also be odd, otherwise there would be a location $\ell'' = (x, y' - 1)$ that precedes ℓ' and satisfies condition 2. But since $y \geqslant y' - 1$ and both y and y' are odd, we need to have $y \geqslant y'$.

From the previous arguments we know that in fact $\ell = (x, y)$ satisfies both conditions 1 and 2. We can thus apply Theorem 13 to the sets $Z_\ell^{\searrow}$ and $Z_\ell^{\nwarrow}$, deriving the existence of two idempotent loops L_1, L_2 and two components C_1, C_2 of L_1, L_2, respectively, such that

- $\max(L_2) < x < \min(L_1)$,
- $\ell_1 \lhd \mathsf{an}(C_1) \lhd \ell \lhd \mathsf{an}(C_2) \lhd \ell_2$,
- $\mathsf{out}(\mathsf{tr}(C_1)), \mathsf{out}(\mathsf{tr}(C_2)) \neq \varepsilon$.

In particular, since $\mathsf{an}(C_1)$ is to the right of $\mathsf{an}(C_2)$ w.r.t. the order of positions, we know that (L_1, C_1, L_2, C_2) is an inversion, and hence $\mathsf{an}(C_1)\ \mathsf{S}^*\ \mathsf{an}(C_2)$. But this contradicts the assumption that $\rho[\ell_1, \ell_2]$ does not overlap with any S^*-block. $\square$

Proposition 31. *Given a functional two-way transducer $\mathcal{T}$, one can construct in* 3EXPTIME *a one-way transducer $\mathcal{T}'$ such that $\mathcal{T}' \subseteq \mathcal{T}$ and $\mathsf{dom}(\mathcal{T}') \supseteq U$.*

Proof. Given an input u, the transducer $\mathcal{T}'$ will guess (and check) a successful run ρ of $\mathcal{T}$ on u, together with a decomposition $(\rho[\ell_i, \ell_{i+1}])_i$ of ρ into blocks and diagonals. The decomposition will be used by $\mathcal{T}'$ to simulate the output of ρ left-to-right, thus proving that $\mathcal{T}' \subseteq \mathcal{T}$. Moreover, $u \in U$ implies the existence of a successful run that can be decomposed, thus proving that $\mathsf{dom}(\mathcal{T}') \supseteq U$. We now provide some details of the construction of $\mathcal{T}'$.

Guessing the run ρ is standard (see, for instance, [5, 17]): it amounts to guess the crossing sequences $\rho|x$ for each position x of the input. Recall that this is a bounded amount of information for each x, since the run is normalized. As concerns the decomposition of ρ, it can be encoded by the endpoints ℓ_i of its factors, that is, by annotating the position of each ℓ_i as the level of ℓ_i. In a similar way $\mathcal{T}'$ guesses the information of whether each factor $\rho[\ell_i, \ell_{i+1}]$ is a diagonal or a block.

Thanks to the definition of decomposition (Def. 23), every two distinct factors span across non-overlapping intervals of positions. This means that each position x is covered by exactly one factor of the decomposition. We call this factor the *active factor at position* x. The mode of computation of the transducer will depend on the type of active factor: if the active factor is a diagonal (resp. a block), then we say that $\mathcal{T}'$ is in *diagonal mode* (resp. *block mode*). Below we describe the behaviour for these two modes of computation.

Diagonal mode. We recall the key condition satisfied by the diagonal $\rho[\ell_i, \ell_{i+1}]$ that is active at position x (cf. Def. 22 and Figure 6): there exists a location $\ell_x = (x, y_x)$ between ℓ_i and ℓ_{i+1} such that the words $\mathsf{out}(\rho \mid Z^{\nwarrow}_{\ell_x})$ and $\mathsf{out}(\rho \mid Z^{\searrow}_{\ell_x})$ have length at most $\boldsymbol{B}$, where $Z^{\nwarrow}_{\ell_x} = [\ell_x, \ell_2] \cap \big([0, x] \times \mathbb{N}\big)$ and $Z^{\searrow}_{\ell_x} = [\ell_1, \ell_x] \cap \big([x, \omega] \times \mathbb{N}\big)$.

Besides the run ρ and the decomposition, the transducer $\mathcal{T}'$ will also guess the locations $\ell_x = (x, y_x)$, that is, will annotate each x with the corresponding y_x. Without loss of generality, we can assume that the function that associates each position x with the guessed location $\ell_x = (x, y_x)$ is monotone, namely, $x \leqslant x'$ implies $\ell_x \trianglelefteq \ell_{x'}$. While the transducer $\mathcal{T}'$ is in diagonal mode, the goal is to preserve the following invariant:

After reaching a position x covered by the active diagonal, $\mathcal{T}'$ must have produced the output of ρ up to location ℓ_x.

To preserve the above invariant when moving from x to the next position $x + 1$, the transducer should output the word $\mathsf{out}(\rho[\ell_x, \ell_{x+1}])$. This word consists of the following parts:

1) The words produced by the single transitions of $\rho[\ell_x, \ell_{x+1}]$ with endpoints in $\{x, x+1\} \times \mathbb{N}$. Note that there are at most h_{max} such words, each of them has length at most c_{max}, and they can all be determined using the crossing sequences at x and $x + 1$ and the information about the levels of ℓ_x and ℓ_{x+1}. We can thus assume that this information is readily available to the transducer.
2) The words produced by the factors of $\rho[\ell_x, \ell_{x+1}]$ that are intercepted by the interval $[0, x]$. Thanks to the definition of diagonal, we know that the total length of these words is at most $\boldsymbol{B}$. These words cannot be determined from the information on $\rho|x$, $\rho|x + 1$, ℓ_x, and ℓ_{x+1} alone, so they need to be constructed while scanning the input. For this, it is important to store additional information.
   More precisely, at each position x of the input, the transducer stores all the outputs produced by the factors of ρ that are intercepted by $[0, x]$ and that occur *after* a location of the form $\ell_{x'}$, for any $x' \geqslant x$ that is covered by a diagonal. This clearly includes the previous words when $x' = x$, but also other words that might be used later for processing other diagonals. Moreover, by exploiting the properties of diagonals, one can prove that those words have length at most $\boldsymbol{B}$, so they can be stored with triply exponentially many states. Using classical techniques, the stored information can be maintained while scanning the input u using the guessed crossing sequences of ρ.
3) The words produced by the factors of $\rho[\ell_x, \ell_{x+1}]$ that are intercepted by the interval $[x + 1, \omega]$. These words must be guessed, since they depend on a portion of the input that has not been processed yet. Accordingly, the guesses need to be stored into memory, so that they can be checked later. Formally, the transducer stores, for each position x, the guessed words that correspond to the outputs produced by the factors of ρ intercepted by $[x, \omega]$ and occurring *before* a location of the form $\ell_{x'}$, for any $x' \leqslant x$ that is covered by a diagonal.

Block mode. Suppose that the active factor $\rho[\ell_i, \ell_{i+1}]$ is a block. Let $I = [x_i, x_{i+1}]$ be the set of positions covered by this factor. Moreover, for each position $x \in I$, let $Z^{\leftarrow}_x = [\ell_i, \ell_{i+1}] \cap \big([0, x] \times \mathbb{N}\big)$ and $Z^{\rightarrow}_x = [\ell_i, \ell_{i+1}] \cap \big([x, \omega] \times \mathbb{N}\big)$. We recall the key property of a block (cf. Definition 22 and Figure 6): the word $\mathsf{out}(\rho[\ell_{i_x}, \ell_{i_x+1}])$ is almost periodic with bound $\boldsymbol{B}$, and the words $\mathsf{out}(\rho \mid Z^{\leftarrow}_{x_i})$ and $\mathsf{out}(\rho \mid Z^{\rightarrow}_{x_{i+1}})$ have length at most $\boldsymbol{B}$.

For the sake of simplicity, suppose that $\mathsf{out}(\rho[\ell_i, \ell_{i+1}]) = w_1\, w_2, w_3$, where w_2 is periodic with period $\boldsymbol{B}$ and w_1, w_2 have length at most $\boldsymbol{B}$. Similarly, let $w_0 = \mathsf{out}(\rho \mid Z^{\leftarrow}_{x_i})$ and $w_4 = \mathsf{out}(\rho \mid Z^{\rightarrow}_{x_{i+1}})$. The invariant preserved by $\mathcal{T}'$ in block mode is the following:

After reaching a position x covered by the active block $\rho[\ell_i, \ell_{i+1}]$, $\mathcal{T}'$ must have produced the output of the prefix of ρ up to location ℓ_i, followed by a prefix of $\mathsf{out}(\rho[\ell_i, \ell_{i+1}]) = w_1\, w_2\, w_3$ of the same length as $\mathsf{out}(\rho \mid Z^{\leftarrow}_x)$.

The initialization of the invariant is done when reaching the left endpoint x_i of the interval I. At this moment, it suffices that $\mathcal{T}'$ outputs a prefix of $w_1\, w_2\, w_3$ of the same length as $w_0 = \mathsf{out}(\rho \mid Z^{\leftarrow}_{x_i})$, thus bounded by $\boldsymbol{B}$. Symmetrically, when reaching the right endpoint x_{i+1} of I, $\mathcal{T}'$ will have produced almost the entire word $\mathsf{out}(\rho[\ell_1, \ell_i])\, w_1\, w_2\, w_3$, but without the suffix of length $|w_4| \leqslant \boldsymbol{B}$. Thus, before moving to the next factor of the decomposition, the transducer will have to produce the remaining suffix, so as to complete the output of ρ up to location ℓ_{i_x+1}.

It remains to describe how the above invariant can be maintained when moving from a position x to the next position $x + 1$ inside I. For this, it is convenient to succinctly represent the word w_2 by its repeating pattern, say v, of length at most $\boldsymbol{B}$. To determine the symbols that have to be output at each step, the transducer will maintain a pointer on either $w_1\, v$ or w_3. The pointer is increased in a deterministic way, and precisely by the amount $\big|\mathsf{out}(\rho \mid Z^{\leftarrow}_{x+1})\big| - \big|\mathsf{out}(\rho \mid Z^{\leftarrow}_x)\big|$. The only exception is when the pointer lies in $w_1\, v$, but its increase would go over $w_1\, v$: in this case the transducer has the choice to either bring the pointer back to the beginning of v (representing a periodic output inside w_2), or move it to w_3. Of course, this is a non-deterministic choice, but it can be validated when reaching the right endpoint of I. Concerning the number of symbols that need to be emitted at each step, this can be determined from the crossing sequences at x and $x+1$, and from the knowledge of the lowest and highest levels of locations that are at position x and between ℓ_i and ℓ_{i+1}. We denote the latter levels by y_x^- and y_x^+, respectively.

Overall, this shows how to maintain the invariant of the block mode, assuming that the levels y_x^-, y_x^+ are known, as well as the words w_0, w_1, v, w_3, w_4 of bounded length. Like the mapping $x \mapsto \ell_x = (x, y_x)$ used in diagonal mode, the mapping $x \mapsto (y_x^-, y_x^+)$ can be guessed and checked using the crossing sequences. Similarly, the words w_1, v, w_3 can be guessed just before entering the active block, and can be checked along the process. As concerns the words w_0, w_4, these can be guessed and checked in a way similar to the words that we used in diagonal mode. More precisely, for each position x of the input, the transducer stores the following additional information:

1) the outputs produced by the factors of ρ that are intercepted by $[0, x]$ and that occur after the beginning ℓ_j of a block, where $\ell_j = (x_j, y_j)$ and $x_j \geqslant x$;
2) the outputs produced by the factors of ρ that are intercepted by $[x, \omega]$ and that occur before the ending ℓ_{j+1} of a block, where $\ell_{j+1} = (x_{j+1}, y_{j+1})$ and $x_{j+1} \leqslant x$.

Thanks to the properties of blocks, the above words have length at most $\boldsymbol{B}$ and can be maintained while processing the input and the crossing sequences. Finally, we observe that the words, together with the information given by the lowest and highest levels y_x^-, y_x^+, for both $x = x_i$ and $x = x_{i+1}$, are sufficient for determining the content of w_0 and w_4.

The above constructions give a one-way transducer $\mathcal{T}'$ of size triple exponential in $\mathcal{T}$. ☐

Theorem 36. *A functional two-way transducer $\mathcal{T}$ is sweeping definable iff it is k-pass sweeping definable, for $k = 2h_{\mathsf{max}} \cdot (2^{3e_{\mathsf{max}}} + 1)$.*

Proof. Suppose that $\mathcal{T}$ is not k-pass sweeping definable for $k = 2h_{\mathsf{max}} \cdot (2^{3e_{\mathsf{max}}} + 1)$. We aim at proving that $\mathcal{T}$ is not m-pass sweeping definable for all $m > 0$. By Theorem 35, we know that there exist a successful run ρ and a k-inversion $\overline{\mathcal{I}} = (\mathcal{I}_0, \dots, \mathcal{I}_{k-1})$ of it, with $\mathcal{I}_i = (L_i, C_i, L'_i, C'_i)$, that is not safe. We consider the locations of ρ that are visited between the beginning of an inversion $\mathcal{I}_i$ and the ending of the next co-inversion $\mathcal{I}_{i+1}$. Formally, for all even indices $i = 0, 2, \dots, k-1$, we let

$$K_i \;=\; \big[\mathsf{an}(C_i), \mathsf{an}(C'_{i+1})\big].$$

We then project each K_i on the x-coordinates:

$$X_i \;=\; \big\{x \;:\; \exists\, \ell = (x, y) \in K_i\big\}.$$

Since K_i is an interval of locations and the transducer $\mathcal{T}$ can only move its head between consecutive positions, we know that each X_i is an interval of positions. Hereafter, we often use the term "interval" to denote a set of the form X_i, for some even index $i \in \{0, 2, \dots, \kappa - 1\}$.

Below we prove that there is a large enough set of pairwise non-overlapping intervals:

Claim. *There is a set $\mathcal{X} = \{X_i\}_{i \in I}$ of cardinality $n = 2^{3e_{\mathsf{max}}} + 1$ such that $X \cap X' = \varnothing$ for all $X \neq X' \in \mathcal{X}$.*

Proof. In this proof, we consider an ordering on the intervals X_i different from the one induced by the indices i. This is given by the lexicographic order on the endpoints, where the dominant element is the rightmost endpoint, namely, we let $X_i < X_j$ if either $\max(X_i) < \max(X_j)$, or $\max(X_i) = \max(X_j)$ and $\min(X_i) < \min(X_j)$.

We construct the set $\mathcal{X}$ inductively, by following the lexicographic ordering. Formally, for all $j = 0, \dots, n$, we construct:

- a set $\mathcal{X}_j$ of size j such that $X \cap X' = \varnothing$ for all $X \neq X' \in \mathcal{X}_j$
- a set $\mathcal{X}'_j$ of size at least $h_{\mathsf{max}} \cdot (2^{3e_{\mathsf{max}}} + 1 - j)$ such that, for all $X \in \mathcal{X}_j$ and all $X' \in \mathcal{X}'_j$, $\max(X) < \min(X')$ (namely, all intervals of $\mathcal{X}'_j$ are strictly to the right of the intervals of $\mathcal{X}_j$).

The base case $j = 0$ of the induction is easy: we let $\mathcal{X}_0 = \varnothing$ and $\mathcal{X}'_0$ be the set of all intervals. It only suffices to observe that $\mathcal{X}'_0$ has cardinality $\frac{k}{2} = h_{\mathsf{max}} \cdot (2^{3e_{\mathsf{max}}} + 1)$.

For the inductive step, suppose that $j < n = 2^{3e_{\mathsf{max}}} + 1$ and that we constructed $\mathcal{X}_j$ and $\mathcal{X}'_j$ satisfying the inductive hypothesis. We let X be the least element in $\mathcal{X}'_j$ according to the lexicographic order (note that $\mathcal{X}'_j \neq \varnothing$ since $j < n$). Accordingly, we define $\mathcal{X}_{j+1} = \mathcal{X}_j \cup \{X\}$ and $\mathcal{X}'_{j+1}$ as the subset of $\mathcal{X}'_j$ that contains the intervals strictly to the right of X. It remains to verify that $\mathcal{X}'_{j+1}$ has cardinality at least $h_{\mathsf{max}} \cdot \big(2^{3e_{\mathsf{max}}} + 1 - (j+1)\big)$. For this we recall that the run ρ is normalized. This implies that there are at most h_{max} intervals in $\mathcal{X}'_j$ that cover the position $x = \max(X)$. All other intervals of $\mathcal{X}'_j$ are necessarily to the right of X: indeed, because X is minimal in the lexicographic ordering, we know that every interval of $\mathcal{X}'_j$ has the right endpoint to the right of x, and as they do not cover the position x, their left endpoint too. This shows that there are at most h_{max} intervals in $\mathcal{X}'_j \setminus \mathcal{X}'_{j+1}$, so $|\mathcal{X}'_{j+1}| \geqslant h_{\mathsf{max}} \cdot \big(2^{3e_{\mathsf{max}}} + 1 - (j+1)\big)$. ▫ *(claim)*

Turning back to the proof of the theorem, we consider the left endpoints of the intervals in $\mathcal{X}$, say

$$\overleftarrow{X} \;=\; \{\min(X) \;:\; X \in \mathcal{X}\}.$$

Since $|\overleftarrow{X}| > 2^{3e_{\mathsf{max}}}$, we can use Theorem 12 to derive the existence of three distinct positions $x < x' < x'' \in \overleftarrow{X}$ such that $[x, x']$ and $[x', x'']$ are consecutive idempotent loops of ρ with the same effect (see also the proof of Theorem 13 for a similar claim). We let $L = [x, x'']$ be the union of those two loops, and we consider the intermediate position x'. We recall that x' is the left endpoint of an interval of $\mathcal{X}$, which we denote by X_i for simplicity. We also recall that X_i is the set of positions visited by a factor of the run ρ that goes from the first anchor $\mathsf{an}(C_i)$ of the inversion $\mathcal{I}_i = (L_i, C_i, L'_i, C'_i)$ to the second anchor $\mathsf{an}(C'_{i+1})$ of the co-inversion $\mathcal{I}_{i+1} = (L_{i+1}, C_{i+1}, L'_{i+1}, C'_{i+1})$.

We claim that the inversion $\mathcal{I}_i$ and the co-inversion $\mathcal{I}_{i+1}$ occur in the same factor intercepted by L. Indeed, the factor $\rho[\mathsf{an}(C_i), \mathsf{an}(C'_{i+1})]$ visits only positions inside the interval X_i. Moreover, the endpoints of X_i are strictly between the endpoints of L, namely,

$$\min(L) \;=\; x \;<\; x' \;=\; \min(X_i) \;\leqslant\; \max(X_i) \;<\; x'' \;=\; \max(L).$$

This shows that the inversion $\mathcal{I}_i = (L_i, C_i, L'_i, C'_i)$ and the co-inversion $\mathcal{I}_{i+1} = (L_{i+1}, C_{i+1}, L'_{i+1}, C'_{i+1})$ occur in the same factor intercepted by L, which we denote by α.

Now, we can easily introduce new copies of the factor α, and hence new copies of the (co)-invesions $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$, by pumping the idempotent loop L. Formally, for all $m > 0$, we denote by $\mathcal{I}_i^{(1)}, \dots, \mathcal{I}_i^{(m)}$ (resp. $\mathcal{I}_{i+1}^{(1)}, \dots, \mathcal{I}_{i+1}^{(m)}$) the m copies of the inversion $\mathcal{I}_i$ (resp. the m copies of the co-inversion $\mathcal{I}_{i+1}$) that appear in the pumped run $\mathsf{pump}_L^m(\rho)$. For the sake of simplicity, we assume that those copies are listed according to their order of occurrence in the pumped run, namely,

$$\mathcal{I}_i^{(1)} \;\lhd\; \mathcal{I}_{i+1}^{(1)} \;\lhd\; \mathcal{I}_i^{(2)} \;\lhd\; \mathcal{I}_{i+1}^{(2)} \;\lhd\; \dots \;\lhd\; \mathcal{I}_i^{(m)} \;\lhd\; \mathcal{I}_{i+1}^{(m)}$$

(the order $\lhd$ is extended from locations to (co-)inversions in the natural way).

Towards a conclusion, we observe that $\big(\mathcal{I}_i^{(1)}, \mathcal{I}_{i+1}^{(1)}, \dots, \mathcal{I}_i^{(m)}, \mathcal{I}_{i+1}^{(m)}\big)$ is a $2m$-inversion of the successful run $\mathsf{pump}_L^m(\rho)$ of $\mathcal{T}$. Moreover, this $2m$-inversion is not safe, since it consists of (co-)inversions that do not generate periodic outputs — more formally, the period of the word $\mathsf{out}(\mathsf{tr}(C_i))\ \mathsf{out}(\rho[\mathsf{an}(C_i), \mathsf{an}(C'_i)])\ \mathsf{out}(\mathsf{tr}(C'_i))$ (resp. $\mathsf{out}(\mathsf{tr}(C_{i+1}))\ \mathsf{out}(\rho[\mathsf{an}(C_{i+1}), \mathsf{an}(C'_{i+1})])\ \mathsf{out}(\mathsf{tr}(C'_{i+1}))$) is larger than $\boldsymbol{B}$ or does not divide $|\mathsf{out}(\mathsf{tr}(C_i))|$ and $|\mathsf{out}(\mathsf{tr}(C'_i))|$ (resp. $|\mathsf{out}(\mathsf{tr}(C_{i+1}))|$ and $|\mathsf{out}(\mathsf{tr}(C'_{i+1}))|$). By Theorem 35, this proves that $\mathcal{T}$ is not m-pass sweeping definable. Finally, since the above holds for all $m > 0$, we conclude that $\mathcal{T}$ is not sweeping definable. □