First-order definability of rational transductions: An algebraic approach

Emmanuel Filiot, Olivier Gauwin, Nathan Lhote

First-order definability of rational transductions: An algebraic approach *

Emmanuel Filiot
Université Libre de Bruxelles
efiliot@ulb.ac.be

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

Nathan Lhote
Université de Bordeaux, LaBRI, CNRS
Université Libre de Bruxelles
nlhote@labri.fr

Abstract

The algebraic theory of rational languages has provided powerful decidability results. Among them, one of the most fundamental is the definability of a rational language in the class of aperiodic languages, *i.e.*, languages recognized by finite automata whose transition relation defines an aperiodic congruence. An important corollary of this result is the first-order definability of monadic second-order formulas over finite words.

Our goal is to extend these results to rational transductions, *i.e.* word functions realized by finite transducers. We take an algebraic approach and consider definability problems of rational transductions in a given variety of congruences (or monoids).

The strength of the algebraic theory of rational languages relies on the existence of a congruence canonically attached to every language, the syntactic congruence. In a similar spirit, Reutenauer and Schützenberger have defined a canonical device for rational transductions, that we extend to establish our main contribution: an effective characterization of $\mathbf{V}$-transductions, *i.e.* rational transductions realizable by transducers whose transition relation defines a congruence in a (decidable) variety $\mathbf{V}$. In particular, it provides an algorithm to decide the definability of a rational transduction by an aperiodic finite transducer.

Using those results, we show that the FO-definability of a rational transduction is decidable, where FO-definable means definable in a first-order restriction of logical transducers *à la Courcelle*.

Categories and Subject Descriptors F.4.2 [*Mathematical Logic and Formal Languages*]: Formal Languages

Keywords rational word transductions, definability problems, first-order logic, algebraic characterizations

1. Introduction

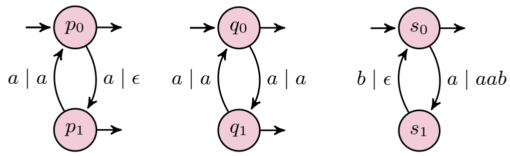
A key aspect of formal language theory is the relationship between logic, automata and algebra, established on a number of structures (*e.g.* words and trees). In this paper, we investigate these connections for rational transductions (*i.e.* word-to-word functions defined by finite transducers), and related definability problems: given a transducer, is its transduction defined by an object of a given class?

Rational languages Rational languages of finite words form a robust class that enjoys computational, logical and algebraic characterizations. They correspond for instance to languages defined by finite automata, monadic second-order logic (MSO), and finite monoids. These powerful connections have also been established for subclasses of rational languages: at the computational level, by imposing structural restrictions on finite automata, at the logical level, by putting restrictions on the use of quantifiers and predicates *e.g.*, and at the algebraic level, by considering varieties (sometimes called pseudo-varieties) of finite monoids. Most notably, first-order definable languages are known to correspond to languages defined by *counter-free* automata, and to languages recognized by *aperiodic* finite monoids (see (Diekert and Gastin 2008) for a survey on first-order definable languages). More generally, a whole theory relating monoid varieties and logical fragments has been established in (Straubing 1994). There is a tight correspondence between the notions of recognizability by monoid and recognizability by congruence, through their quotient. In this paper, for convenience, we choose the congruence view. Roughly speaking, a congruence variety is a class of congruences of finite index with reasonable closure properties: *e.g.* the classes of congruences of finite index, aperiodic congruences, commutative congruences are congruence varieties.

A powerful tool in this context is the existence of a canonical minimal deterministic automaton for each rational language, whose states are equivalence classes of the so called *right syntactic congruence* (or Myhill-Nerode congruence), a congruence canonically attached to each language. The right syntactic congruence $\sim_L$ of a language L has a strong property with respect to varieties $\mathbf{V}$: If L is recognized by some $\mathbf{V}$-automaton, *i.e.* a finite automaton whose transition relation defines a congruence in $\mathbf{V}$, called the transition congruence, then $\sim_L$ belongs to $\mathbf{V}$. In other words, the minimal deterministic automaton that recognizes L is a $\mathbf{V}$-automaton. This well-known result provides a way to decide whether an automaton $\mathcal{A}$ is equivalent to some $\mathbf{V}$-automaton, as long as $\mathbf{V}$ is a decidable variety, *i.e.* a variety with decidable membership problem. It suffices to construct the minimal deterministic automaton for the language recognized by $\mathcal{A}$ and check whether its transition congruence belongs to $\mathbf{V}$. As an application, it yields for instance the decidability of FO in MSO on finite words: given an MSO sentence φ, it is decidable whether φ is equivalent to some FO formula. More generally, fragments of MSO that are characterized by decidable varieties have, through automata minimization, a decidable definability problem in MSO (Straubing 1994).

Our goal is to extend these decidability results, considered as jewels of theoretical computer science, to rational transductions.

* This work was partially supported by the *ExStream* project (ANR-13-JS02-0010), the ARC project *Transform* (French speaking community of Belgium) and the Belgian FNRS PDR project *Flare*. Emmanuel Filiot is research associate at F.R.S.-FNRS.



(a) $a^n \mapsto a^{\lfloor \frac{n}{2} \rfloor}$ (b) $a^n \mapsto a^n$ (c) $(ab)^n \mapsto (aab)^n$

Figure 1. Finite state transducers.

Rational transductions Transductions are functions from (finite) words to words. Rational transductions are the transductions realized by finite state transducers, *i.e.* automata extended with outputs. Figure 1 depicts three examples of transducers. Initial states are states with source-free incoming arrow, and final states are states with a target-free outgoing arrow. In transducer 1(a), both p_0 and p_1 are final but only p_0 is initial. When reading a sequence of a, the left transducer outputs a half the time, and the empty word ϵ the other half. This transducer maps any word of the form a^n to $a^{\lfloor \frac{n}{2} \rfloor}$, for all $n \geq 0$. Transducer 1(b) simply realizes the identity function on a^n for all $n \geq 0$, and transducer 1(c) maps any word of the form $(ab)^n$ to $(aab)^n$, and the transduction is undefined for other input words.

Transition congruence of a transducer To define the transition congruence $\approx_\mathcal{T}$ of a transducer $\mathcal{T}$ over an alphabet Σ, one only considers the input symbols: it is the *transition congruence* of its underlying input automaton (obtained by erasing the output). Informally, a word u defines a binary relation on the states of $\mathcal{T}$ – (p, q) are in the relation if there is a run from p to q on u– and two words are equivalent if they define the same relation. For the transducer 1(a), *e.g.*, the word a is equivalent to aaa: both send state p_0 to p_1 and state p_1 to p_0. There are in fact two equivalence classes, that of ϵ (equal to $\{\epsilon, aa, a^4, \dots\}$) and that of a (equal to $\{a, a^3, a^5, \dots\}$).

Contributions A transducer is aperiodic if its transition congruence $\approx_\mathcal{T}$ is aperiodic, *i.e.* $u^n \approx_\mathcal{T} u^{n+1}$ for all words $u \in \Sigma^*$ and all n larger than some bound that depends only on $\mathcal{T}$. Transducer 1(c) is aperiodic, but not transducers 1(a) and 1(b).

Despite their non-aperiodicity, are transducers 1(a) and 1(b) equivalent to some aperiodic transducer? In this paper, one objective is to automatize this question, for the class of aperiodic congruences but more generally for any decidable variety $\mathbf{V}$ of congruences. Due to non-determinism, an input word can have possibly many output words by a transducer. In this paper, we consider the class of *unambiguous transducers*, which defines exactly the class of rational transductions (Eilenberg 1974). In other words, one wants to decide whether a rational function, given by a transducer, realizes a $\mathbf{V}$-transduction, *i.e.* a transduction definable by some unambiguous transducer whose transition congruence belongs to $\mathbf{V}$ (we call it a $\mathbf{V}$-transducer). As our first main result, we answer this question positively:

Theorem 1. *Let $\mathbf{V}$ be a decidable variety. It is decidable whether a given transducer realizes a $\mathbf{V}$-transduction.*

While it is clear that transducer 1(b) is equivalent to a single state aperiodic transducer, it turns out that transducer 1(a) is not, although its domain, the set of words of the form a^n for all $n \geq 0$, is an aperiodic language. As a matter of facts, the tools needed to prove Theorem 1 are not simple extensions of what is known from the theory of languages. In general, there is no minimal transducer for rational transductions. However, for a subclass of transductions, the *sequential transductions*, minimization exists.

Sequential transductions are the rational transductions realized by sequential transducers, the subclass of transducers whose underlying input automaton is deterministic. There exists a minimal sequential transducer for any sequential transduction f (Choffrut 2003), based on a congruence relation introduced by Choffrut, that takes into account the output words. We show that this congruence belongs to a variety $\mathbf{V}$ if and only if f is realized by a sequential $\mathbf{V}$-transducer.

For rational transductions in general, there is no unique minimal transducer. However, canonicity is available for another device, *bimachines* (Schützenberger 1961; Eilenberg 1974). Roughly speaking, a bimachine is a sequential transducer with a deterministic regular look-ahead. It consists of a left deterministic automaton, reading words from left to right, and a deterministic right automaton (the look-ahead), reading words from right to left. The output words are then computed based on the information given by the two automata. If both the left and right automata are $\mathbf{V}$-automata, then the bimachine they define is called a $\mathbf{V}$-bimachine. Given a rational function f, represented as a transducer, (Reutenauer and Schützenberger 1991) have shown how to compute a canonical bimachine for f. Unfortunately, even if f is $\mathbf{V}$-rational, this canonical bimachine may not be a $\mathbf{V}$-bimachine. However, we show how to canonically attach to f a finite and computable set of bimachines, among which one is a $\mathbf{V}$-bimachine iff f is a $\mathbf{V}$-transduction. This addresses the conjecture stated in (Reutenauer and Schützenberger 1991), Section 5.3, that "there are only a finite number of minimal bimachines computing (a function) α".

On the logical side, we take Courcelle's logical transducers as a formalism to define word transductions (Courcelle 1994), based on first-order logic over the predicates for labels, and the natural order between positions. Intuitively, an FO-transducer is a finite set of FO formulas that define the predicates of the output word, interpreted on a bounded number of copies of the input word structure. If one takes MSO instead of FO, it is known that MSO-transducers correspond to transductions definable with deterministic two-way transducers (Engelfriet and Hoogeboom 2001), and therefore are much beyond the expressiveness of rational transductions. They can for instance copy the input word, or swap unbounded subwords of it. To capture rational transductions, MSO-transducers have been restricted to their *order-preserving* variant (Bojanczyk 2014; Filiot 2015). We consider the same restriction on FO-transducers, that we call $\mathsf{FO}_{\mathsf{op}}$-transducers. We show that aperiodic transductions are exactly the transductions realized by $\mathsf{FO}_{\mathsf{op}}$-transducers, and, as a consequence of Theorem 1, we obtain our second result:

Theorem 2. *It is decidable whether a rational transduction (as a finite transducer) is $\mathsf{FO}_{\mathsf{op}}$-definable.*

Related work In (Schützenberger 1965), Schützenberger proved that languages definable by an aperiodic deterministic finite-state automaton are exactly star-free languages (see (Diekert and Kufleitner 2015) for a simplified version). Star-free languages were proved to be captured by first-order logic (with the order predicate) in (McNaughton and Papert 1971). This provides a way to decide first-order definability of a regular language, through aperiodicity (see *e.g.* (Diekert and Gastin 2008)).

For word functions, a recent work (Cadilhac et al. 2015) shows that it is decidable whether a deterministic rational transduction is definable in the circuit class AC^0 (resp. ACC^0), also using an algebraic approach. Other results relate transduction classes, but without definability procedures. We briefly present some of them, and refer the reader to (Filiot 2015) for a more complete picture.

One of these relations has been established in (Lautemann et al. 1999): Aperiodic functional non-deterministic length-preserving transducers capture length-preserving FO-transducers. Here, aperiodicity of a *non-deterministic* transducer is defined as the aperiodicity of the deterministic automaton obtained after subset construction on the input automaton of the transducer. In (McKenzie et al. 2006) this result is generalized to group varieties (beyond aperiodic monoids), and also to two-way transducers and bimachines, but always on length-preserving functions.

Streaming string transducers and two-way transducers both capture MSO-transducers. Aperiodic streaming string transducers capture FO-translations (Filiot et al. 2014), where aperiodicity applies on the transition monoid and the variable dependencies. Aperiodic two-way transducers also capture FO-transducers (Carton and Dartois 2015). Here aperiodicity only applies to the transition monoid.

Another non-effective characterization of aperiodic rational transductions is established in (Reutenauer and Schützenberger 1995), by relating the period of any rational language with the period of its inverse image by the transduction.

Organization Section 2 gathers the definitions of languages, transductions and related concepts used in this paper. Section 3 is devoted to algebraic characterizations of transductions, and Section 4 to applications in logics. Hence Theorem 1 is proved in Section 3, and Theorem 2 in Section 4. Finally, Section 5 contains discussions on these results and related open problems.

2. Rational languages and transductions

2.1 Rational languages

Words and languages An alphabet Σ is a finite set. A word over Σ is an element of the free monoid Σ^*. We denote by ϵ the empty word. For $w \in \Sigma^*$, $|w|$ denotes its length. In particular $|\epsilon| = 0$. The set of *positions* of w is $\mathrm{pos}(w) = \{1, \dots, |w|\}$, and in particular, $\mathrm{pos}(\epsilon) = \varnothing$. For a word w and $i, j \in \mathrm{pos}(w)$ such that $i \leq j$, $w[i]$ denotes the i-th letter of w, $w[i{:}j]$ the factor of w from position i to position j (both included), and we set $w[{:}i] = w[1{:}i]$ and $w[i{:}] = w[i{:}|w|]$. The *prefix order* $\preceq$ on words is defined by $u \preceq v$ if $u = \epsilon$, or $u = v[{:}i]$ for some $i \in \mathrm{pos}(v)$ (if it exists). If $u \preceq v$, we denote by $u^{-1}v$ the word v' such that $v = uv'$. By $u \wedge v$ we denote the *longest common prefix* of any two words u and v, and by $\|u, v\|$ the value $|u| + |v| - 2|u \wedge v|$. It is well-known that $\|., .\|$ defines a distance. Finally, a *language* $L \subseteq \Sigma^*$ is a set of words, and the *prefix* of L is the set $P(L)$ of all prefixes of words of L.

Finite automata A finite automaton (or just automaton for short) over an alphabet Σ is a tuple $\mathcal{A} = (Q, I, F, \Delta)$ where Q is a finite set of states, $I \subseteq Q$ (resp. $F \subseteq Q$) is a set of initial (resp. final) states, and $\Delta \subseteq Q \times \Sigma \times Q$ is a transition relation. $\mathcal{A}$ is *deterministic* if I is a singleton and for any two rules $(p, \sigma, q_1), (p, \sigma, q_2) \in \Delta$, it holds that $q_1 = q_2$. A *run* r of an automaton $\mathcal{A} = (Q, I, F, \Delta)$ on a word $w \in \Sigma^*$ of length n is a word $r = q_0 \dots q_n$ over Q such that $(q_i, w[i+1], q_{i+1}) \in \Delta$ for all $i \in \{0, \dots, n-1\}$. It is accepting if $q_0 \in I$ and $q_n \in F$. A word w is accepted by $\mathcal{A}$ if there exists an accepting run of $\mathcal{A}$ over it. The *language recognized by* $\mathcal{A}$ is the set $[\![\mathcal{A}]\!]$ of words accepted by $\mathcal{A}$. We often write $p \xrightarrow{w}_{\mathcal{A}} q$ (or simply $p \xrightarrow{w} q$) whenever there exists a run r on w such that $r[1] = p$ and $r[|r|] = q$. A state q is *accessible* if there is a word w and an initial state q_0 such that $q_0 \xrightarrow{w} q$, and an automaton is *accessible* if all its states are accessible. If for any word there exists at most one accepting run on $\mathcal{A}$, we say that $\mathcal{A}$ is *unambiguous*. Finally, the class of *rational languages* over Σ is defined as the class of languages recognized by finite automata.

Congruences and recognizability Let Σ be an alphabet and let $\sim$ be an equivalence relation on Σ^*. We say that $\sim$ is a *right congruence* if it satisfies $u \sim v \Rightarrow u\sigma \sim v\sigma$ for all $u, v \in \Sigma^*, \sigma \in \Sigma$. Symmetrically we define a *left congruence* ($u \sim v \Rightarrow \sigma u \sim \sigma v$), and a *congruence* is defined as both a left and a right congruence. For $u \in \Sigma^*$, we denote by $[u]_\sim$ (or $[u]$ if clear from the context) its equivalence class, and by $\Sigma^*/_\sim$ the quotient of Σ^* by $\sim$, *i.e.* $\Sigma^*/_\sim = \{[u]_\sim \mid u \in \Sigma^*\}$. We say that $\sim$ has *finite index* if $\Sigma^*/_\sim$ is finite. Concatenation naturally extends to congruence classes as follows: for all $u, v \in \Sigma^*$, $[u]_\sim[v]_\sim = [uv]_\sim$. Since $\sim$ is a congruence, the latter is well-defined as it does not depend on the choices of u and v. With this operation, $\Sigma^*/_\sim$ forms a monoid whose neutral element is $[\epsilon]_\sim$.

Central examples of congruences in this paper are the *syntactic congruence* $\equiv_L$ of a language L and the *transition congruence* $\approx_{\mathcal{A}}$ of an automaton $\mathcal{A}$, defined as follows:

$$\begin{array}{lllllll} u & \equiv_L & v & \Leftrightarrow & (\forall x, y \in \Sigma^*, & xuy \in L & \Leftrightarrow \quad xvy \in L) \\ u & \approx_{\mathcal{A}} & v & \Leftrightarrow & (\forall p, q \in Q, & p \xrightarrow{u}_{\mathcal{A}} q & \Leftrightarrow \quad p \xrightarrow{v}_{\mathcal{A}} q) \end{array}$$

We say that a language L is recognized by a congruence $\sim$ if there exists $P \subseteq \Sigma^*/_\sim$ such that $L = \{u \in \Sigma^* \mid [u] \in P\}$. For example, by taking $P = L/_{\equiv_L}$, one can see that L is recognized by $\equiv_L$. It is also well-known that a language L is rational iff it is recognized by a congruence of finite index.

Another useful example is the *right transition congruence* $\sim_{\mathcal{A}}$ of a deterministic automaton $\mathcal{A}$, defined as follows:

$$u \quad \sim_{\mathcal{A}} \quad v \quad \Leftrightarrow \quad (\forall q \in Q, \quad q_0 \xrightarrow{u}_{\mathcal{A}} q \quad \Leftrightarrow \quad q_0 \xrightarrow{v}_{\mathcal{A}} q)$$

Let $\sim_1$ and $\sim_2$ be two equivalence relations on Σ^*. We say that $\sim_1$ is *finer* than $\sim_2$, or that $\sim_2$ is *coarser* than $\sim_1$, denoted by $\sim_1 \sqsubseteq \sim_2$ if for all $u, v \in \Sigma^*$ we have $u \sim_1 v \Rightarrow u \sim_2 v$. For instance, $\equiv_L$ is the coarsest congruence recognizing L, for any language L. If $\sim_1$ and $\sim_2$ are congruences of finite index then $\sim_1 \sqcap \sim_2$ (seen as the set intersection) is a congruence of finite index, finer than both $\sim_1$ and $\sim_2$. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two deterministic automata, we will say by extension that $\mathcal{A}_1$ is *finer* than $\mathcal{A}_2$ if we have $\sim_{\mathcal{A}_1}$ finer than $\sim_{\mathcal{A}_2}$, and we write $\mathcal{A}_1 \sqsubseteq \mathcal{A}_2$. We consider right congruences because they are naturally equivalent to (accessible) deterministic automata.

2.2 Algebraic characterization of classes of regular languages

Congruence varieties We say that a set $\mathbf{V}$ of congruences of finite index is a *congruence variety* (variety for short) if it is closed under intersection of congruences (on the same alphabet) and the taking of coarser congruences. Let $\mathbf{V}$ be a variety and let Σ be an alphabet. An automaton with transition letters in Σ is a $\mathbf{V}$-automaton if its transition congruence is in $\mathbf{V}$. A language over Σ is a $\mathbf{V}$-language if it is recognized by a $\mathbf{V}$-automaton. Since a variety is stable by taking coarser congruences, a language is a $\mathbf{V}$-language if and only if its syntactic congruence is in $\mathbf{V}$. We denote by $\mathcal{L}(\mathbf{V})$ the set of $\mathbf{V}$-languages.

Remark 1. The notion of recognizability by a congruence of finite index $\sim$ is equivalent to the already studied notion of recognizability by a stamp (*i.e.* a morphism from a free monoid to a finite monoid) $\varphi : \Sigma^* \to M$ by setting $u \sim v$ iff $\varphi(u) = \varphi(v)$ for one direction, and by taking $M = \Sigma^*/_\sim$ and $\varphi : u \mapsto [u]_\sim$ for the other direction. In this paper we have chosen the congruence approach for simplicity reasons. Furthermore, the notion of congruence variety we define here is more general than the one of $\mathcal{C}$-variety of (Pin and Straubing 2005), indeed a $\mathcal{C}$-variety of stamps is in particular a congruence variety. We make this choice simply because our results still hold in this more general framework.

Definability problem Given a class of languages Λ, the Λ-definability problem asks, given an automaton recognizing a language L whether $L \in \Lambda$.

Decidable variety Let Σ be an alphabet. A congruence of finite index over Σ can be given as a morphism $\varphi : \Sigma^* \to M$, with M a finite monoid, by taking $u \sim v$ iff $\varphi(u) = \varphi(v)$. The morphism φ can itself be given explicitly by a function $\phi : \Sigma \to M$. Let $\mathbf{V}$ be a variety. The *membership problem* for $\mathbf{V}$ asks, for a congruence of finite index, given explicitly, whether it is in $\mathbf{V}$. In particular the $\mathcal{L}(\mathbf{V})$-definability problem reduces to the membership problem for $\mathbf{V}$. We say that $\mathbf{V}$ is *decidable* if its membership problem is decidable.

Example 1. Let $\mathbf{A}$ denote the variety of *aperiodic* congruences, *i.e.* of congruences $\sim$ verifying: $\exists n > 0 \quad \forall w \in \Sigma^* \quad w^n \sim w^{n+1}$. This property is stable by intersection and taking coarser congruences hence $\mathbf{A}$ is indeed a variety. The $\mathbf{A}$-definability problem is PSPACE-complete (Cho and Huynh 1991).

2.3 Rational transductions

Transductions A *transduction* over a finite alphabet Σ is a partial function of $\Sigma^* \to \Sigma^*$. In the literature a transduction is, generally speaking, a binary relation over Σ^*, however in this paper we only consider functional relations. We denote by $\mathrm{dom}(f)$ the domain of a transduction f.

Finite transducers A finite transducer[1] (or just transducer for short) over an alphabet Σ is a tuple $\mathcal{T} = (\mathcal{A}, o, i, t)$ where $\mathcal{A} = (Q, I, F, \Delta)$ is a finite automaton, $o : \Delta \to \Sigma^*$ is the *output function*, $i : I \to \Sigma^*$ is the *initial function* and $t : F \to \Sigma^*$ is the *final function*. Let $r = q_0 \dots q_n$ be a run of $\mathcal{A}$ on the word u. We write $q_0 \xrightarrow{u|v}_{\mathcal{T}} q_n$ (or simply $q_0 \xrightarrow{u|v} q_n$) whenever $q_0 \xrightarrow{u}_{\mathcal{A}} q_n$ and $v = o(q_0, u[0], q_1) \dots o(q_{n-1}, u[n], q_n)$. If r is an accepting run and $w = i(q_0) v\, t(q_n)$ then we say that (u, w) is *realized* by $\mathcal{T}$. Then the *relation realized* by $\mathcal{T}$ is the set of pairs of words realized by $\mathcal{T}$ and denoted by $[\![\mathcal{T}]\!]$. The transducer $\mathcal{T}$ is called *unambiguous* (resp. *sequential*) if $\mathcal{A}$ is unambiguous (resp. deterministic). In both cases $[\![\mathcal{T}]\!]$ is a transduction and we denote $(u, w) \in [\![\mathcal{T}]\!]$ by $[\![\mathcal{T}]\!](u) = w$. The class of *rational transductions* (resp. *sequential transductions*) is defined as the class of transductions realized by unambiguous (resp. sequential) finite transducers.

Remark 2. It is decidable in polynomial time whether a transducer realizes a transduction (*i.e.* is functional) and in that case there exists an unambiguous transducer realizing the same transduction (see *e.g.* (Berstel and Boasson 1979)).

$\mathbf{V}$*-transductions* Let $\mathbf{V}$ be a variety. A finite transducer is called a $\mathbf{V}$*-transducer* if its automaton is a $\mathbf{V}$-automaton. A transduction is called $\mathbf{V}$*-rational* (resp. $\mathbf{V}$*-sequential*) if it is realized by an unambiguous (resp. sequential) $\mathbf{V}$-transducer.

3. Algebraic characterizations of transductions

In this section we establish algebraic characterizations, for a given variety $\mathbf{V}$, of $\mathbf{V}$-sequential and $\mathbf{V}$-rational transductions based on notions of canonical congruences associated with a given transduction. These characterizations are effective whenever $\mathbf{V}$ is decidable.

In the first subsection we show that, for a given variety $\mathbf{V}$, the minimal sequential transducer defined in (Choffrut 2003) is a $\mathbf{V}$-transducer if and only if its transduction is definable by a sequential $\mathbf{V}$-transducer, which provides a way to decide if a sequential transducer realizes a $\mathbf{V}$-sequential transduction.

In the other two subsections, we deal with the more difficult case of rational transductions. Refining the methods of (Reutenauer and Schützenberger 1991), we obtain this time not one minimal transducer but a finite set of transducers such that, for a given variety $\mathbf{V}$, one of these transducers is a $\mathbf{V}$-transducer if and only if the transduction is realizable by some unambiguous $\mathbf{V}$-transducer. Thus we obtain a way to decide if a transducer realizes a $\mathbf{V}$-rational transduction, as stated in Theorem 1.

[1] This type of transducer is sometimes called *real-time* (Sakarovitch 2009). In the general case, a transition of a transducer may be labelled by any word, however such a transducer is always equivalent to a real-time one.

3.1 Sequential Transductions

In order to decide if a given rational language belongs to some variety, one only has to consider the syntactic congruence of the language which is coarser than any congruence recognizing the language. In the same spirit, sequential transductions can be minimized by writing the outputs as soon as possible and the resulting minimal transducer has the coarsest congruence of any sequential transducer realizing the transduction. We describe the construction given in (Choffrut 2003) of this minimal transducer and show that its transition congruence is indeed the coarsest. For a proof of correctness of the construction we refer the reader to the original paper.

Minimization Let $\mathcal{T} = (\mathcal{A}, o, i, t)$ be a sequential transducer with $\mathcal{A} = (Q, I, F, \Delta)$, and let $f = [\![\mathcal{T}]\!]$. Let us define the transducer $\mathcal{T}_f = (\mathcal{A}_f, o_f, i_f, t_f)$ which depends only on f. The construction of $\mathcal{T}_f$ is based on a notion of syntactic congruence for f, that we now define. Intuitively, two words u and v are equivalent if they are equivalent for the Myhill-Nerode right congruence of $\mathrm{dom}(f)$, and if for any suffix w such that both uw and vw are in $\mathrm{dom}(f)$, the outputs $f(uw)$ and $f(vw)$ have a common suffix which only depends on w. In other words, the effect of u and v on the translation of w is the same.

Formally, $\widehat{f}$ is defined on $P(\mathrm{dom}(f))$ by: $\widehat{f}(u) = \wedge\{f(uw) \mid uw \in \mathrm{dom}(f)\}$ for $u \in P(\mathrm{dom}(f))$. The *syntactic congruence* of f is denoted by $\sim_f$ and defined as: for $u, v \in \Sigma^*$, $u \sim_f v$ if for all $w \in \Sigma^*$, $uw \in \mathrm{dom}(f) \Leftrightarrow vw \in \mathrm{dom}(f)$ and if $u, v \in \mathrm{dom}(\widehat{f})$, $\widehat{f}(u)^{-1} f(uw) = \widehat{f}(v)^{-1} f(vw)$. Then we define:

- $Q_f = \{[u] \mid u \in \mathrm{dom}(\widehat{f})\}$
- $I_f = \{[\epsilon]\}$ (we assume that the domain of f is non empty)
- $F_f = \{[u] \mid u \in \mathrm{dom}(f)\}$
- $\Delta_f = \{([u], \sigma, [u\sigma]) \mid u\sigma \in \mathrm{dom}(\widehat{f})\}$
- $o_f([u], \sigma, [u\sigma]) = \widehat{f}(u)^{-1} \widehat{f}(u\sigma)$
- $i_f([\epsilon]) = \widehat{f}(\varepsilon)$
- $t_f([u]) = \widehat{f}(u)^{-1} f(u)$

Theorem 3. *Let $\mathbf{V}$ be a decidable variety. Given a sequential transduction $f : \Sigma^* \to \Sigma^*$ (as a transducer $\mathcal{T}$), it is decidable whether f is $\mathbf{V}$-sequential.*

Before proving the theorem we consider the following property.

Proposition 1. *Let $\mathbf{V}$ be a variety, let $\mathcal{A}_1$, $\mathcal{A}_2$ be two deterministic automata such that $\mathcal{A}_1 \sqsubseteq \mathcal{A}_2$ and $\mathcal{A}_2$ is accessible. If $\mathcal{A}_1$ is a $\mathbf{V}$-automaton, then $\mathcal{A}_2$ is a $\mathbf{V}$-automaton.*

Proof. Let $\mathcal{A}_1$, $\mathcal{A}_2$ be two deterministic automata such that $\mathcal{A}_2$ is accessible. We only need to show that $\approx_{\mathcal{A}_1} \sqsubseteq \approx_{\mathcal{A}_2}$ since any variety is stable by the taking of coarser congruences. Let us assume that $\sim_{\mathcal{A}_1} \sqsubseteq \sim_{\mathcal{A}_2}$ and let $u \approx_{\mathcal{A}_1} v$. Then, for any word w, we have that $wu \sim_{\mathcal{A}_1} wv$, hence for any w, $wu \sim_{\mathcal{A}_2} wv$. Since $\mathcal{A}_2$ is accessible, we have that $u \approx_{\mathcal{A}_2} v$. Hence, if $\mathcal{A}_1$ is a $\mathbf{V}$-automaton, then $\mathcal{A}_2$ is a $\mathbf{V}$-automaton. □

Proof of Theorem 3. Let us show that $\mathcal{T}_f$ is a $\mathbf{V}$-sequential transducer iff there exists a $\mathbf{V}$-sequential transducer realizing f. This will entail the result because $\mathcal{T}_f$ is effectively computable from any sequential transducer realizing f (Choffrut 2003). Let $\mathcal{T} = (\mathcal{A}, o, i, t)$ be a $\mathbf{V}$-transducer realizing f. We want to show that $\mathcal{A}_f$ is a $\mathbf{V}$-automaton, since by definition $\mathcal{A}_f$ is accessible, it suffices to show according to Proposition 1 that $\mathcal{A} \sqsubseteq \mathcal{A}_f$. Let $u \sim_{\mathcal{A}} v$ and let us show that $u \sim_f v$. Let w be a word such that $p \xrightarrow{u}_{\mathcal{A}} q \xrightarrow{w}_{\mathcal{A}} r$ such that $p \in I$, $r \in F$. This implies that $p \xrightarrow{v}_{\mathcal{A}} q \xrightarrow{w}_{\mathcal{A}} r$. Hence we have that for all $w \in \Sigma^*$, $uw \in \mathrm{dom}(f) \Leftrightarrow vw \in \mathrm{dom}(f)$. Let

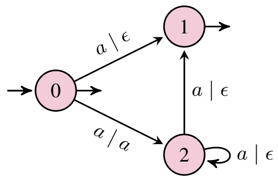


Figure 2. $\mathbf{I}$-transducer.

w be a word such that $p \xrightarrow{u|x} q \xrightarrow{w|y} r$ and $p \xrightarrow{v|x'} q \xrightarrow{w|y} r$ with $p \in I$, $r \in F$. Let us define f_q the transduction realized by $\mathcal{T}_q = ((Q, q, F, \Delta), o, q \mapsto \epsilon, t)$. We have $\widehat{f}(u) = i(p)xx_q$ and $\widehat{f}(v) = i(p)x'x_q$ with $x_q = \widehat{f}_q(\epsilon)$. We also have $f(uw) = i(p)xyt(r)$ and $f(vw) = i(p)x'yt(r)$. Hence we have $\widehat{f}(u)^{-1}f(uw) = x_q^{-1}yt(r) = \widehat{f}(v)^{-1}f(vw)$ which means that $u \sim_f v$ and concludes the proof. $\square$

Remark 3. In general, for a given variety $\mathbf{V}$, a sequential transduction that is $\mathbf{V}$-rational may not be necessarily $\mathbf{V}$-sequential, and we give an example of this below.

Let $\mathbf{I}$ be the variety of idempotent congruences, *i.e.* for $\sim$ in $\mathbf{I}$, $w \in \Sigma^*$ we have $w \sim w^2$. In Figure 2 we show a sequential function which is $\mathbf{I}$-rational yet not $\mathbf{I}$-sequential. The transduction f realized is obviously sequential. Let us show by contradiction that it cannot be realized by a sequential $\mathbf{I}$-transducer. Let us assume there exists a sequential $\mathbf{I}$-transducer realizing f. We have $p \xrightarrow{a|\epsilon} q$ with $p \in I$ and $q \in F$. We also have $i(p) = t(q) = \epsilon$. Since $f(aa) = a$ and $a \sim aa$, we must have $q \xrightarrow{a|a} q$, and then we obtain $f(aaa) = aa$ which is a contradiction.

It is important however to note that in the particular case of $\mathbf{A}$, a sequential transduction that is $\mathbf{A}$-rational is necessarily $\mathbf{A}$-sequential. The proof of the following proposition is quite technical and is given in the Appendix.

Proposition 2. *A sequential transduction is* $\mathbf{A}$*-sequential if and only if it is* $\mathbf{A}$*-rational.*

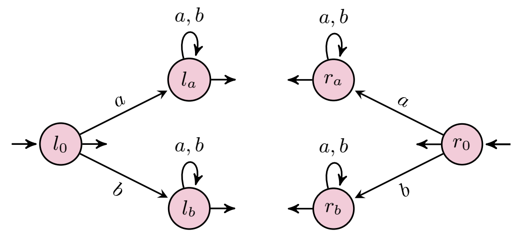
3.2 Rational Transductions

For rational transductions, $\mathbf{V}$-rationality is also decidable, as stated in Theorem 1. This section is devoted to proving this result.

Since rational transductions cannot, in general, be realized by a sequential transducer, a unique congruence is not enough to characterize a transduction. Let us consider the example of the transduction $f : w\sigma \mapsto \sigma w$, with $\sigma \in \Sigma$ and $w \in \Sigma^*$. This transduction is not sequential (if Σ has more than one letter), and if we apply the construction of (Choffrut 2003) we obtain a right congruence with infinite index, because in order to obtain the image of $w\sigma$ in a sequential fashion, one has to remember the whole word w before outputting σ. In this case, by adding a look-ahead information about the last letter of the input, one can output the image of $w\sigma$ sequentially. It is the main idea of (Reutenauer and Schützenberger 1991) to get a canonical object for rational transductions. They prove indeed that for rational transductions, a finite amount of look-ahead information is sufficient to realize the transduction in a sequential manner, knowing that information.

In particular, in (Reutenauer and Schützenberger 1991), it is shown that rational transductions can be characterized by pairs of congruences: a left congruence which gives look-ahead information on the suffix of the word and a Choffrut-like right congruence which depends on the left one. Based on those congruences, they define a computational model called bimachines.

Bimachines A bimachine is a model of computation introduced by (Schützenberger 1961) and shown to be equivalent to (functional) transducers. A bimachine is composed of two automata, a left automaton, which is just a deterministic automaton, and a right automaton. A right automaton is an automaton reading words from right to left deterministically. A bimachine can be seen as a transducer with look-ahead where the look-ahead at some position of the word is given by the state of the right automaton.



(a) Left automaton $\mathcal{L}$ (b) Right automaton $\mathcal{R}$

Figure 3. Automata of bimachine $\mathcal{B}$.

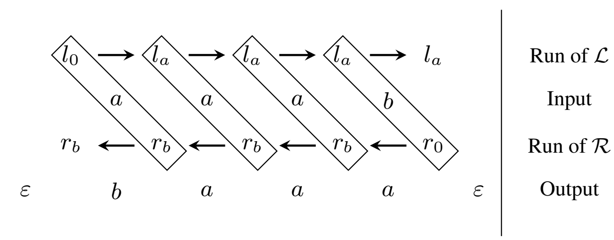


Figure 4. Execution of bimachine $\mathcal{B}$.

The runs of a right automaton $\mathcal{R}$ are defined as the runs of a left one. Let r be a run on a right automaton, it is accepting if $r[1]$ is final and $r[|r|]$ is initial. We write $s_2 \xleftarrow{w}_{\mathcal{R}} s_1$ whenever there is a run r of $\mathcal{R}$ on w such that $r[1] = s_2$ and $r[|r|] = s_1$. Furthermore, the transitions of a right automaton are deterministic, *i.e.* it has only one initial state and for any two transitions $(p_1, \sigma, q), (p_2, \sigma, q)$ it holds that $p_1 = p_2$. The *transition congruence* $\approx_{\mathcal{R}}$ of a right automaton $\mathcal{R}$ with a set of states R is defined as $u \approx_{\mathcal{R}} v \Leftrightarrow (\forall p, q \in R, p \xleftarrow{u}_{\mathcal{R}} q \Leftrightarrow p \xleftarrow{v}_{\mathcal{R}} q)$.

A *bimachine* over an alphabet Σ is a tuple $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \omega, \lambda, \rho)$ where $\mathcal{L} = (L, \{l_0\}, F_{\mathcal{L}}, \Delta_{\mathcal{L}})$ is a deterministic left automaton (or just left automaton), $\mathcal{R} = (R, \{r_0\}, F_{\mathcal{R}}, \Delta_{\mathcal{R}})$ is a deterministic right automaton (or right automaton), $\omega : L \times \Sigma \times R \to \Sigma^*$ is the *output function*, $\lambda : F_{\mathcal{R}} \to \Sigma^*$ is the *left final function* and $\rho : F_{\mathcal{L}} \to \Sigma^*$ is the *right final function*. Let u be a word such that the runs of $\mathcal{L}$ and of $\mathcal{R}$ on u, $l = l_0 \dots l_n$ and $r = r_n \dots r_0$ are both accepting. We write

$$[\![\mathcal{B}]\!](u) = \lambda(r_n)\omega(l_0, u[1], r_{n-1}) \dots \omega(l_{n-1}, u[n], r_0)\rho(l_n)$$

and we say that $[\![\mathcal{B}]\!]$ is the transduction realized by $\mathcal{B}$.

Example 2. In Figure 3 we give the automata of a bimachine $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \omega, \lambda, \rho)$ realizing the transduction $f_{\text{swap}} : \sigma w \tau \mapsto \tau w \sigma$ for $\sigma, \tau \in \Sigma$, $w \in \Sigma^*$ and $\Sigma = \{a, b\}$ and maps words of length less than two to themselves. For $l \in L, r \in R$ and $\sigma \in \Sigma$ we define:

$$\lambda(r) = \rho(l) = \epsilon \text{ and } \omega(l, \sigma, r) = \begin{cases} \sigma \text{ if } l \neq l_0 \text{ and } r \neq r_0 \\ \sigma \text{ if } l = l_0 \text{ and } r = r_0 \\ \tau \text{ if } l = l_\tau \text{ and } r = r_0 \\ \tau \text{ if } l = l_0 \text{ and } r = r_\tau \end{cases}$$

Figure 4 illustrates the execution of $\mathcal{B}$ on the word $aaab$.

Left transition congruence For a given right automaton $\mathcal{R} = (R, \{r_0\}, F_{\mathcal{R}}, \Delta_{\mathcal{R}})$, we define the *left transition congruence* $\sim_{\mathcal{R}}$ as $u \sim_{\mathcal{R}} v \Leftrightarrow (\forall r \in R, r \xleftarrow{u}_{\mathcal{R}} r_0 \Leftrightarrow r \xleftarrow{u}_{\mathcal{R}} r_0)$. As for left automata, we say that $\mathcal{R}_1$ is finer than $\mathcal{R}_2$ or write $\mathcal{R}_1 \sqsubseteq$

$\mathcal{R}_2$ if $\sim_{\mathcal{R}_1} \sqsubseteq \sim_{\mathcal{R}_2}$. Notice that for simplicity we use the same notations for left and right automata and we rely on the context to differentiate between them.

Remark 4. One can easily see that a bimachine with a trivial right automaton (one state, accepts all words) is equivalent to a sequential transducer.

Canonical bimachine The construction of a canonical bimachine associated with a rational transduction was given in (Reutenauer and Schützenberger 1991). The idea is to refine the minimization of sequential transducers using what can be seen as a look-ahead. For a fixed right automaton, a minimal left automaton can be defined, like for sequential transducers, by producing the outputs as early as possible, given the look-ahead information of the right automaton. A canonical right automaton is defined in (Reutenauer and Schützenberger 1991), which gives a construction of a completely canonical machine. Let us define this canonical machine.

Let f be a rational transduction on Σ. The *left congruence* of f is defined by $\forall u, v \in \Sigma^*$, $u \leftharpoonup_f v$ if:

- $\forall w \in \Sigma^*, wu \in \mathrm{dom}(f) \Leftrightarrow wv \in \mathrm{dom}(f)$.
- $\sup\{\|f(wu), f(wv)\| \mid wu, wv \in \mathrm{dom}(f)\} < \infty$

The finiteness of the index of $\leftharpoonup_f$ is a powerful characterization of rational transductions and we refer the interested reader to (Reutenauer and Schützenberger 1991).

We define the canonical right automaton $\mathcal{R}_f = (R, \{r_0\}, F, \Delta)$. For the rest of this section $[w]$ denotes the class of w in $\Sigma^*/_{\leftharpoonup_f}$. We take $R = \Sigma^*/_{\leftharpoonup_f}$, $r_0 = [\epsilon]$, $F = \{[w] \mid w \in \mathrm{dom}(f)\}$ and $\Delta = \{([\sigma w], \sigma, [w]) \mid \sigma \in \Sigma, w \in \Sigma^*\}$.

Minimal left automaton Let f be a rational transduction and let $\mathcal{R}$ be a right automaton finer than $\mathcal{R}_f$. Our goal here is to construct a canonical bimachine $\mathcal{B}_f(\mathcal{R}) = (\mathit{Left}_f(\mathcal{R}), \mathcal{R}, \omega, \lambda, \rho)$, which realizes f, as described in (Reutenauer and Schützenberger 1991) but we refer the reader to the original paper for a proof of correctness. We first give the construction of a canonical left automaton $\mathit{Left}_f(\mathcal{R})$ (or simply $\mathit{Left}(\mathcal{R})$ when it is clear from the context). For simplicity, we will write $[w]_{\mathcal{R}}$ instead of $[w]_{\sim_{\mathcal{R}}}$ for any word $w \in \Sigma^*$. Let w be a word and let $u \in P(\mathrm{dom}(f))$, we define $\widehat{f}_{[w]_{\mathcal{R}}}(u) = \wedge\{f(uv) \mid v \in [w]_{\mathcal{R}}\}$ and if $u \notin P(\mathrm{dom}(f))$, $\widehat{f}_{[w]_{\mathcal{R}}}$ is not defined. This word is the longest possible output upon reading u, knowing that the suffix is in $[w]_{\mathcal{R}}$. To define the left automaton $\mathit{Left}_f(\mathcal{R})$, we need a right congruence: $u \sim_L v$ if for any letter σ, any $w, z \in \Sigma^*$ we have:

- $uz \in \mathrm{dom}(f) \;\Leftrightarrow\; vz \in \mathrm{dom}(f)$
- $\widehat{f}_{[\epsilon]_{\mathcal{R}}}(uz)^{-1} f(uz) = \widehat{f}_{[\epsilon]_{\mathcal{R}}}(vz)^{-1} f(vz)$, if $uz, vz \in \mathrm{dom}(f)$
- $\widehat{f}_{[\sigma w]_{\mathcal{R}}}(uz)^{-1} \widehat{f}_{[w]_{\mathcal{R}}}(uz\sigma) = \widehat{f}_{[\sigma w]_{\mathcal{R}}}(vz)^{-1} \widehat{f}_{[w]_{\mathcal{R}}}(vz\sigma)$

Intuitively, congruence classes of $\sim_L$ will be the states of $\mathit{Left}_f(\mathcal{R})$. Then, the second line ensures that for the states reached after reading uz and vz, the final output is the same. The third line states that the output of a transition reading σ is the same for the states reached after reading uz and vz.

From $\sim_L$ we define the automaton $\mathit{Left}_f(\mathcal{R}) = (L, \{l_0\}, F, \Delta)$ where $L = \Sigma^*/_{\sim_L}$, $l_0 = [\epsilon]_{\sim_L}$, $F = \{[w]_{\sim_L} \mid w \in \mathrm{dom}(f)\}$ and $\Delta = \{([w]_{\sim_L}, \sigma, [w\sigma]_{\sim_L}) \mid \sigma \in \Sigma, w \in \Sigma^*\}$.

Now that we have the automata of the bimachine, the output functions are defined naturally:

- $\omega([u]_{\sim_L}, \sigma, [v]_{\mathcal{R}}) = \widehat{f}_{[\sigma v]_{\mathcal{R}}}(u)^{-1} \widehat{f}_{[v]_{\mathcal{R}}}(u\sigma)$
- $\lambda([v]_{\mathcal{R}}) = \widehat{f}_{[v]_{\mathcal{R}}}(\epsilon)$
- $\rho([u]_{\sim_L}) = \widehat{f}_{[\epsilon]_{\mathcal{R}}}(u)^{-1} f(u)$

Theorem 4. *(Reutenauer and Schützenberger 1991) Let f be a rational transduction. Let $\mathcal{R}$ be a right automaton finer than $\mathcal{R}_f$. Then the bimachine $\mathcal{B}_f(\mathcal{R}) = (\mathit{Left}_f(\mathcal{R}), \mathcal{R}, \omega, \lambda, \rho)$ realizes f.*

Remark 5. The important result of (Reutenauer and Schützenberger 1991) is that the bimachine $\mathcal{B}_f(\mathcal{R}_f)$ is completely canonical *i.e.* does not depend on any description of f, and is computable. We can define symmetrically the *right congruence* of f by $u \rightharpoonup_f v$ if $\forall w$, $uw \in \mathrm{dom}(f) \Leftrightarrow vw \in \mathrm{dom}(f)$ and $\sup\{\|f(uw), f(vw)\| \mid uw, vw \in \mathrm{dom}(f)\} < \infty$. We can then define $\mathcal{L}_f = (L, \{l_0\}, F, \Delta)$ by $L = \Sigma^*/_{\rightharpoonup_f}$, $l_0 = [\epsilon]_{\rightharpoonup_f}$, $F = \{[w]_{\rightharpoonup_f} \mid w \in \mathrm{dom}(f)\}$ and $\Delta = \{([w]_{\rightharpoonup_f}, \sigma, [w\sigma]_{\rightharpoonup_f}) \mid \sigma \in \Sigma, w \in \Sigma^*\}$. For $\mathcal{L}$ finer than $\mathcal{L}_f$ one can also define symmetrically $\mathit{Right}_f(\mathcal{L})$ and $\mathcal{B}_f(\mathcal{L})$ with $\mathcal{L}$ and $\mathit{Right}_f(\mathcal{L})$ as its left and right automata. Note also that for a total sequential transduction f, the automaton $\mathcal{R}_f$ is trivial and $\mathcal{B}_f(\mathcal{R}_f)$ is exactly the minimal transducer of (Choffrut 2003).

3.3 V-rational transductions

In this section we naturally define $\mathbf{V}$-bimachines as bimachines with $\mathbf{V}$-automata and show they exactly capture $\mathbf{V}$-rational transductions. Since we have, thanks to (Reutenauer and Schützenberger 1991), a canonical bimachine one could hope that it suffices to compute this canonical device to decide if a transduction is $\mathbf{V}$-rational. Unfortunately the canonical bimachine is not necessarily a $\mathbf{V}$-bimachine even for a $\mathbf{V}$-rational transduction. The case of the transduction given in Figure 2 is a good example of this: It is $\mathbf{I}$-rational, however the canonical bimachine is not. Indeed, since the outputs of this transduction f are of bounded length, all words are congruent with respect to $\leftharpoonup_f$. This means that the right automaton of $\mathcal{R}_f$ is trivial, which is consistent with the fact that f is sequential (and total). However, as we have seen in Remark 3, f is not $\mathbf{I}$-sequential hence the canonical bimachine cannot be an $\mathbf{I}$-bimachine. We have to refine the construction of (Reutenauer and Schützenberger 1991), and we show that a rational transduction admits not one but a finite set of minimal (*i.e.* pairwise incomparable) bimachines, in the strong sense that if there exists a $\mathbf{V}$-bimachine realizing a transduction then one of the minimal bimachines is a $\mathbf{V}$-bimachine. However we only get this result for total functions, but we show that the $\mathbf{V}$-rationality problem for partial functions can easily be reduced to the problem for total functions.

Proposition 3. *Let $\mathbf{V}$ be a variety. A transduction is $\mathbf{V}$-rational if and only if it is realized by a $\mathbf{V}$-bimachine.*

Proof. Let f be a transduction realized by $\mathcal{T} = (\mathcal{A}, o, i, t)$ an unambiguous $\mathbf{V}$-transducer. We will define a $\mathbf{V}$-bimachine $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \omega, \lambda, \rho)$ realizing the same transduction. We consider the congruence $\approx_{\mathcal{A}}$ which is by definition in $\mathbf{V}$. Then we define $\mathcal{L} = (L, \{l_0\}, F_{\mathcal{L}}, \Delta_{\mathcal{L}})$ with: $L = \Sigma^*/_{\approx_{\mathcal{A}}}$, $l_0 = [\epsilon]$, $F_{\mathcal{L}} = \{[w] \mid w \in \mathrm{dom}(f)\}$ and $\Delta_{\mathcal{L}} = \{([w], \sigma, [w\sigma]) \mid \exists v,\ w\sigma v \in \mathrm{dom}(f)\}$.

Similarly we define $\mathcal{R} = (R, \{r_0\}, F_{\mathcal{R}}, \Delta_{\mathcal{R}})$ with: $R = \Sigma^*/_{\approx_{\mathcal{A}}}$, $r_0 = [\epsilon]$, $F_{\mathcal{R}} = \{[w] \mid w \in \mathrm{dom}(f)\}$ and $\Delta_{\mathcal{R}} = \{([\sigma w], \sigma, [w]) \mid \exists v,\ v\sigma w \in \mathrm{dom}(f)\}$.

Let $([u], \sigma, [v]) \in L \times \Sigma \times R$. If $u\sigma v \in \mathrm{dom}(f)$ then there exists a unique tuple (by unambiguity) $(p_1, p_2, q_1, q_2) \in I \times Q \times Q \times F$ such that $p_1 \xrightarrow{u} p_2$, $(p_2, \sigma, q_1) \in \Delta$ and $q_1 \xrightarrow{v} q_2$. In that case we set $\omega([u], \sigma, [v]) = o(p_2, \sigma, q_1)$. Otherwise the value of $\omega([u], \sigma, [v])$ does not matter and can be set to ϵ for instance. If $u \in \mathrm{dom}(f)$, we define $\lambda([u]) = i(p)$ with p being the first state of the unique accepting run on u which is, again, well defined. It is well defined since any word equivalent to u has an accepting run beginning with p. Similarly, $\rho([u]) = t(p)$ with p being the last state of the unique accepting run on u.

$\mathcal{B}$ is by definition a $\mathbf{V}$-bimachine which is by construction equivalent to $\mathcal{T}$. For a word of this domain, one can check that the two transductions coincide.

Let f be a transduction realized by $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \omega, \lambda, \rho)$ a $\mathbf{V}$-bimachine. We will define an unambiguous $\mathbf{V}$-transducer $\mathcal{T} = (\mathcal{A}, o, i, t)$ realizing the same transduction. This transducer simultaneously simulates the runs of $\mathcal{L}$ and the runs of $\mathcal{R}$ but backwards. We define $\mathcal{A} = (Q, I, F, \Delta)$ with: $Q = L \times R$, $I = \{l_0\} \times F_{\mathcal{R}}$, $F = F_{\mathcal{L}} \times \{r_0\}$ and

$$\Delta = \left\{ ((l_1, r_1), \sigma, (l_2, r_2)) \mid \begin{array}{l} (l_1, \sigma, l_2) \in \Delta_{\mathcal{L}} \\ (r_1, \sigma, r_2) \in \Delta_{\mathcal{R}} \end{array} \right\}$$

It is easily shown that $\approx_{\mathcal{A}}$ is coarser than $\approx_{\mathcal{L}} \sqcap \approx_{\mathcal{R}}$, hence, since $\mathbf{V}$ is a variety it is closed under these operations, $\mathcal{A}$ is a $\mathbf{V}$-automaton. We define $o((l_1, r_1), \sigma, (l_2, r_2)) = \omega(l_1, \sigma, r_2)$, $i(l_0, r) = \lambda(r)$ and $t(l, r_0) = \rho(l)$. $\mathcal{T}$ is an unambiguous $\mathbf{V}$-transducer which is by construction equivalent to $\mathcal{B}$. $\square$

Partial functions Let f be a rational transduction over Σ. The *completion* of f, denoted by $\overline{f}$ is defined by:

$$\begin{array}{rccl} \overline{f}: & \Sigma^* & \rightarrow & (\Sigma \uplus \{\perp\})^* \\ & w & \mapsto & \left\{ \begin{array}{ll} f(w) & \text{if } w \in \mathrm{dom}(f) \\ \perp & \text{otherwise} \end{array} \right. \end{array}$$

where $\perp \notin \Sigma$ is a fresh alphabet symbol.

Proposition 4. *Let Σ be an alphabet, let $\mathbf{V}$ be a variety. Then we have an equivalence:*

- *f is a $\mathbf{V}$-rational transduction.*
- *$\overline{f}$ is a $\mathbf{V}$-rational transduction and dom(f) is a $\mathbf{V}$-language.*

Proof. Let $\mathcal{T}$ be an unambiguous $\mathbf{V}$-transducer realizing f. Since f is $\mathbf{V}$-rational, its domain is a $\mathbf{V}$-language and thus we can define $\mathcal{A}'$ a deterministic automaton recognizing $\Sigma^* \setminus \mathrm{dom}(f)$. This can be done because modifying the initial and final states of an automaton does not affect the transition congruence. Hence we can define $\mathcal{T}'$ an unambiguous $\mathbf{V}$-transducer that outputs $\perp$ on any word not in $\mathrm{dom}(f)$ and rejects otherwise. The union of $\mathcal{T}$ and $\mathcal{T}'$ is unambiguous since their domains do not intersect. The transition congruence of the obtained transducer is coarser than the intersection of the transition congruences of $\mathcal{T}$ and $\mathcal{T}'$, and since $\mathbf{V}$ is a variety it is closed under these operations, we obtain an unambiguous $\mathbf{V}$-transducer realizing $\overline{f}$.

Conversely let $\mathcal{T}$ be an unambiguous $\mathbf{V}$-transducer realizing $\overline{f}$. Since the domain of f is a $\mathbf{V}$-language, we can construct an unambiguous $\mathbf{V}$-transducer behaving as $\mathcal{T}$ but rejecting any word outside of $\mathrm{dom}(f)$: Let $\mathcal{A}$ be the automaton of $\mathcal{T}$ and let $\mathcal{A}'$ be an unambiguous $\mathbf{V}$-automaton recognizing $\mathrm{dom}(f)$. We construct the transducer whose automaton is the product of $\mathcal{A}$ and $\mathcal{A}'$ and behaves as $\mathcal{T}$ but only accepts when $\mathcal{A}'$ reaches an accepting state. The transition congruence of the product is coarser than the intersection of the congruences of the automata, hence,since $\mathbf{V}$ is a variety, we obtain an unambiguous $\mathbf{V}$-automaton realizing f. $\square$

Towards a finite set of automata Let us now give an overview of the proof of Theorem 1, for a total transduction f. The main idea is the following: If f is $\mathbf{V}$-rational, realized by some $\mathbf{V}$-bimachine $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \omega, \lambda, \rho)$, then there exists a bimachine $\mathcal{B}' = (\mathcal{L}', \mathcal{R}', \omega', \lambda', \rho')$ that realizes f, of which we can bound the granularity of the left and right automata by canonical and computable automata, and such that $\mathcal{L} \sqsubseteq \mathcal{L}'$ and $\mathcal{R} \sqsubseteq \mathcal{R}'$. In particular, we can construct $\mathcal{L}'$ such that it satisfies $\mathit{Left}(\mathcal{R}_f) \sqsubseteq \mathcal{L}' \sqsubseteq \mathcal{L}_f$, and $\mathcal{R}'$ such that it satisfies $\mathit{Right}(\mathcal{L}_f) \sqsubseteq \mathcal{R}' \sqsubseteq \mathcal{R}_f$. There is only a finite number of automata $\mathcal{L}'$ and $\mathcal{R}'$ in these intervals. Moreover, since we take them such that $\mathcal{L} \sqsubseteq \mathcal{L}'$ and $\mathcal{R} \sqsubseteq \mathcal{R}'$, and since by assumption $\mathcal{L}$ and $\mathcal{R}$ are $\mathbf{V}$-automata, so are $\mathcal{L}'$ and $\mathcal{R}'$. In other words, $\mathcal{B}'$ is a $\mathbf{V}$-bimachine.

We prove that it suffices to take $\mathcal{L}' = \mathit{Left}(\mathcal{R})$ and $\mathcal{R}' = \mathit{Right}(\mathit{Left}(\mathcal{R}))$. The relations between these automata are depicted on Figure 5. The main objective of the rest of this section is to establish the relations depicted on this figure.

First, we establish the upper bounds:

Proposition 5. *Let f be a total transduction, and let $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \omega, \lambda, \rho)$ be a bimachine realizing f. Then $\mathcal{L} \sqsubseteq \mathcal{L}_f$ and $\mathcal{R} \sqsubseteq \mathcal{R}_f$.*

Proof. Let f be a total transduction and let $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \omega, \lambda, \rho)$ be a bimachine realizing f. We will show that $\mathcal{R}$ is finer than $\mathcal{R}_f$ and the proof for $\mathcal{L} \sqsubseteq \mathcal{L}_f$ is exactly symmetrical. Let $u \sim_{\mathcal{R}} v$, we want to show that $u \leftharpoonup_f v$. We have obviously for any word w, that $wu \in \mathrm{dom}(f) \Leftrightarrow wu \in \mathrm{dom}(f)$ since f is total. Let w be a word and let $l = l_0 \ldots l_{|wu|}$ and $r = r_{|wu|} \ldots r_0$ the runs (they exist since f is total) on wu of $\mathcal{L}$ and $\mathcal{R}$, respectively. Similarly, let $l' = l'_0 \ldots l'_{|wv|}$ and $r' = r'_{|wv|} \ldots r'_0$ the runs on wv of $\mathcal{L}$ and $\mathcal{R}$, respectively. Since $u \sim_{\mathcal{R}} v$ and $\mathcal{R}$ is deterministic, the ends (from right to left) of the runs r and r' are identical, *i.e.* for $0 \leq i \leq |w|$ we have $r_{i+|u|} = r'_{i+|v|}$. Hence for $x = f(wu) \wedge f(wv)$ we have

$$x^{-1}f(wu) = \omega(l_{|w|}, u[1], r_{|u|-1}) \ldots \omega(l_{|wu|-1}, u[|u|], r_0)\rho(l_{|wu|})$$
$$x^{-1}f(wv) = \omega(l_{|w|}, v[1], r_{|v|-1}) \ldots \omega(l_{|wv|-1}, v[|v|], r_0)\rho(l_{|wv|})$$

Finally we obtain $\| f(wu), f(wv) \| \leq k(|u| + |v| + 2)$ where k is the maximum length of any word in the ranges of ω, λ and ρ, which proves that $u \leftharpoonup_f v$. $\square$

To establish the lower bounds, we first state a useful result from (Reutenauer and Schützenberger 1991) for total transductions, which gives us a minimality result for the automaton $\mathit{Left}(\mathcal{R})$ when the right automaton $\mathcal{R}$ is fixed.

Proposition 6. *(Reutenauer and Schützenberger 1991) Let f be a total transduction. Let $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \omega, \lambda, \rho)$ be a bimachine realizing f. Then $\mathcal{L} \sqsubseteq \mathit{Left}(\mathcal{R})$ and $\mathcal{R} \sqsubseteq \mathit{Right}(\mathcal{L})$.*

From this proposition, we derive an important corollary which will give us the lower bounds:

Corollary 1. *Let f be a total transduction and $\mathcal{R} \sqsubseteq \mathcal{R}_f$ a right automaton. Then $\mathit{Left}(\mathcal{R}_f) \sqsubseteq \mathit{Left}(\mathcal{R})$. Symmetrically, if $\mathcal{L} \sqsubseteq \mathcal{L}_f$ then $\mathit{Right}(\mathcal{L}_f) \sqsubseteq \mathit{Right}(\mathcal{L})$.*

Proof. Let f be a total transduction and $\mathcal{R} \sqsubseteq \mathcal{R}_f$. We can assume that $\mathcal{R}$ is accessible and still have $\mathcal{R} \sqsubseteq \mathcal{R}_f$. According to Theorem 4, there is a bimachine realizing f with $\mathit{Left}(\mathcal{R}_f)$ and $\mathcal{R}_f$ as its automata. This means that there is a bimachine with $\mathit{Left}(\mathcal{R}_f)$ and $\mathcal{R}$ as its automata which realizes f, by discarding the extra information given by the finer automaton $\mathcal{R}$ in the following way: We have a well-defined function

$$\begin{array}{rccl} \pi: & \Sigma^*/_{\sim_{\mathcal{R}}} & \rightarrow & \Sigma^*/_{\leftharpoonup_f} \\ & [w]_{\sim_{\mathcal{R}}} & \mapsto & [w] \end{array}$$

Since $\mathcal{R}$ is accessible, we also have a function:

$$\begin{array}{rccl} \alpha: & R & \rightarrow & \Sigma^*/_{\sim_{\mathcal{R}}} \\ & r & \mapsto & [w]_{\sim_{\mathcal{R}}} \text{ such that } r \xleftarrow{w}_{\mathcal{R}} r_0 \end{array}$$

Hence we have the function $\pi \circ \alpha : R \rightarrow \Sigma^*/_{\leftharpoonup_f}$ such that for any word w if $r \xleftarrow{w}_{\mathcal{R}} r_0$ then $\pi \circ \alpha(r) = [w]$, hence the behaviour of $\mathcal{R}_f$ can be simulated by $\mathcal{R}$.

Then according to Proposition 6, $\mathit{Left}(\mathcal{R}_f) \sqsubseteq \mathit{Left}(\mathcal{R})$. The symmetric part of the result is shown in the exact same way. $\square$

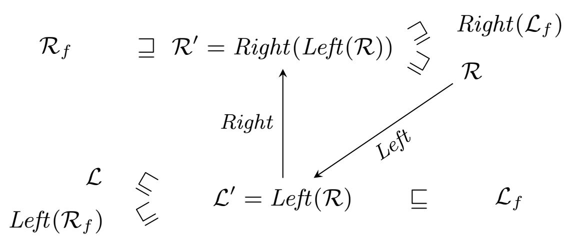


Figure 5. Relations between automata

We are now able to establish the relations of Figure 5, as stated by the following lemma, which gives a sufficient and necessary condition for a total transduction to be $\mathbf{V}$-rational.

Lemma 1. *Let f be a total transduction. f is a $\mathbf{V}$-rational transduction if and only if there exists an automaton $\mathcal{L}'$, with $Left(\mathcal{R}_f) \sqsubseteq \mathcal{L}' \sqsubseteq \mathcal{L}_f$ such that $\mathcal{B}_f(\mathcal{L}')$ is a $\mathbf{V}$-bimachine.*

Proof. If there exists a $\mathbf{V}$-automaton $\mathcal{L}'$ coarser than $Left(\mathcal{R}_f)$ such that $Right(\mathcal{L}')$ is a $\mathbf{V}$-automaton, then f is clearly $\mathbf{V}$-rational.

Conversely, let us assume that f is realized by a $\mathbf{V}$-bimachine $\mathcal{B} = (\mathcal{L}, \mathcal{R}, \omega, \lambda, \rho)$. According to Theorem 4 there is a bimachine realizing f with $Left(\mathcal{R}_f)$ and $\mathcal{R}_f$ as its automata. According to Proposition 6 we know that $\mathcal{L} \sqsubseteq Left(\mathcal{R})$ which means that $Left(\mathcal{R})$ is in $\mathbf{V}$. By Proposition 5, $\mathcal{R}$ is finer than $\mathcal{R}_f$, we also have according to Corollary 1 that $Left(\mathcal{R}_f) \sqsubseteq Left(\mathcal{R})$. From Theorem 4 and Proposition 6, we have that $\mathcal{R} \sqsubseteq Right(Left(\mathcal{R}))$ hence $Right(Left(\mathcal{R}))$ is a $\mathbf{V}$-automaton and $\mathcal{B}_f(Left(\mathcal{R}))$ is a $\mathbf{V}$-bimachine which concludes the proof. $\square$

We have proven the key lemma towards the procedure to decide if a transduction is $\mathbf{V}$-rational. Since there are only a finite number of left automaton coarser than a given left automaton, we are able to define a canonical finite set of bimachines, such that a transduction is $\mathbf{V}$-rational if and only if one of those is a $\mathbf{V}$-bimachine.

Proof of Theorem 1 Let $\mathcal{T}$ be a transducer and $\mathbf{V}$ a decidable congruence variety. First we check in PTime whether $\mathcal{T}$ defines a transduction, *i.e.* is functional (Gurari and Ibarra 1983). Second, we check if $\mathrm{dom}(f)$ is a $\mathbf{V}$-language. If it is not then f is not $\mathbf{V}$-rational. If it is then we construct an unambiguous transducer realizing $\overline{f}$ using the construction given in Proposition 4.

Then we construct the right automaton $\mathcal{R}_{\bar{f}}$ and the left automaton $Left_{\bar{f}}(\mathcal{R}_{\bar{f}})$. They are computable from $\mathcal{T}$ as shown in (Reutenauer and Schützenberger 1991). Then we compute all the $\mathbf{V}$-automata $\mathcal{L}'$ coarser than $Left_{\bar{f}}(\mathcal{R}_{\bar{f}})$, of which there is a finite number, and check if $Right_{\bar{f}}(\mathcal{L}')$ is a $\mathbf{V}$-automaton. According to Lemma 1, if there does not exist such a pair of $\mathbf{V}$-automata, then f is not $\mathbf{V}$-rational. Otherwise the bimachine $\mathcal{B}_{\bar{f}}(\mathcal{L}')$ is a $\mathbf{V}$-bimachine realizing $\overline{f}$.

From Proposition 3 we obtain an unambiguous $\mathbf{V}$-transducer realizing $\overline{f}$. Finally, from Proposition 4 we can construct an unambiguous $\mathbf{V}$-transducer realizing f.

4. First-order definability problems for transductions

The theory of rational languages is rich with results linking automata, logics and algebra, and there have already been successful attempts to lift some of these results to transductions: a monadic second-order logic based transducer model introduced by Courcelle (MSO-transducers) has been shown to be equivalent to deterministic two-way transducers (Engelfriet and Hoogeboom 2001) and, recently, to a one-way deterministic model with registers (Alur and Cerný 2010). More recently, equivalences have been shown between FO-transducers and deterministic two-way transducers with an aperiodic transition monoid (Carton and Dartois 2015).

However, the expressiveness of MSO-transducers, which are equivalent to deterministic two-way transducers, lies way above that of rational transductions. Indeed, two-way transducers can define transductions that do not preserve the order between input symbols: *e.g.*, they can mirror an input word, or copy it twice. Therefore, in this paper, we consider the *order-preserving* restriction of MSO-transducers, which we call $\mathsf{MSO}_{\mathsf{op}}$-transducers, and is known to capture exactly the class of rational transductions (Bojanczyk 2014; Filiot 2015). The first-order fragment of $\mathsf{MSO}_{\mathsf{op}}$-transducers is defined as $\mathsf{MSO}_{\mathsf{op}}$-transducers but with first-order formulas instead of second-order ones.

We show decidability of the following problem: given a rational transduction, is it definable by some $\mathsf{FO}_{\mathsf{op}}$-transducer? In particular, we prove that a transduction is realizable by an $\mathsf{FO}_{\mathsf{op}}$-transducer iff it is $\mathbf{A}$-rational, and we use Theorem 1 to decide $\mathbf{A}$-rationality. Since our translations are effective, we get the decidability of the following definability problem: given an $\mathsf{MSO}_{\mathsf{op}}$-transducer, is it equivalent to some $\mathsf{FO}_{\mathsf{op}}$-transducer ?

Our result relies on the correspondence between languages defined in the variety of aperiodic congruences, and first-order definable languages. Similar correspondences exist for other varieties and logical fragments of MSO (Straubing 1994), and as seen in the previous section, $\mathbf{V}$-rationality of transductions is decidable for all decidable varieties $\mathbf{V}$. This raises the question of whether one could get decidability of the definability of an $\mathsf{MSO}_{\mathsf{op}}$-transducer by an $\mathcal{F}_{\mathsf{op}}$-transducer, for logical fragments $\mathcal{F}$ other than FO. We discuss this question at the end of this section.

4.1 Monadic second-order and first-order logics on words

We first recall the definition of MSO and FO.

Words as logical structures To express properties of words in logics, one sees a word w over an alphabet Σ as a *logical structure*[2] $\tilde{w}$ over the signature $\Xi_\Sigma = \{(\sigma(x))_{\sigma\in\Sigma}, x \preceq y\}$ where $\sigma(x)$ is a unary predicate interpreted as the positions of a word labelled by $\sigma \in \Sigma$, and $\preceq$ is a binary predicate interpreted as the order of the word positions. The *domain* of the structure $\tilde{w}$ is $\mathrm{dom}(w)$, the set of positions of w. From now on, we write w instead of $\tilde{w}$ when it is clear from the context.

Monadic second-order and first-order logics *Monadic second-order formulas* (MSO-formulas) over Ξ_Σ are defined over a countable set of first-order variables $x, y, \ldots$ and a countable set of second-order variables $X, Y, \ldots$ by the following grammar:

$$\phi ::= \exists X\ \phi \mid \exists x\ \phi \mid (\phi \wedge \phi) \mid \neg\phi \mid x \in X \mid \sigma(x) \mid x \preceq y$$

Universal quantifier and other boolean connectives are defined as usual: $\forall x\ \phi := \neg\exists x\ \neg\phi$, $\forall X\ \phi := \neg\exists X\ \neg\phi$, $(\phi_1 \vee \phi_2) := \neg(\neg\phi_1 \wedge \neg\phi_2)$, $(\phi_1 \to \phi_2) := \neg\phi_1 \vee \phi_2$. We also define the formulas $\top$ and $\bot$ as being respectively always and never satisfied. *First-order formulas* (FO-formulas) are the MSO-formulas in which no second-order variable occurs. We do not define the semantics of MSO, nor the standard notion of free and bound variables, but rather refer the reader to (Ebbinghaus and Flum 1995) or (Straubing 1994) for formal definitions. We recall that a closed formulas (or sentence), is a formula without free variables.

Let ϕ be an MSO formula without free second-order variables, we write $\phi(x_1, \ldots, x_m)$ to denote that the free first-order variables of ϕ are $x_1, \ldots, x_m$. Given an MSO formula $\phi(x_1, \ldots, x_m)$ and

[2] For a definition of logical structures see for instance (Ebbinghaus and Flum 1995) or (Straubing 1994).

$w \in \Sigma^*$, we write $w \models \phi(i_1, \ldots, i_m)$ to denote that w, together with the interpretations of x_j by i_j, $j \in \{1, \ldots, m\}$ satisfies ϕ.

The *language defined* by a closed MSO-formula (or FO-formula) ϕ is the set $[\![\phi]\!] = \{w \mid w \models \phi\}$.

Example 3. As a (well-known) example, we show that the language $(ab)^*$ is FO-definable. It suffices to express that the first position is labeled a, the last one is labeled b, and for any position labeled a, the next one is labeled b, and the next next one is labeled a. First, we define the formula for first, last, and successor:

$$\begin{array}{ll} x \prec y \equiv x \preceq y \wedge x \neq y & \text{first}(x) \equiv \neg\exists y, y \prec x \\ \text{last}(x) \equiv \neg\exists y, x \prec y & S(x,y) \equiv x \prec y \wedge \neg\exists z, x \prec z \prec y \end{array}$$

Then, the formula defining $(ab)^*$ is:

$$\begin{array}{l} \forall x,\ (\text{first}(x) \rightarrow a(x)) \wedge (\text{last}(x) \rightarrow b(x)) \wedge \\ \quad \forall y \forall z [a(x) \wedge S(x,y) \rightarrow (b(y) \wedge (S(y,z) \rightarrow a(z)))] \end{array}$$

4.2 Logical transducers

Logical transducers have been defined in (Courcelle 1994; Courcelle and Engelfriet 2012) as a logical way of defining transductions of arbitrary structures. We cast this definition to word transductions. Intuitively, the domain of the output word is defined from a fixed number k of copies of the input word. Then the predicates of the output word structure are defined as formulas with free first-order variables (one for unary label predicates and two for the binary order predicate) interpreted over the input word. In particular, a closed formula ϕ_{dom} defines the domain of the transduction. For all $\sigma \in \Sigma$ and all copies c, a formula $\phi_\sigma^c(x)$ defines the label of the c-th copy of the input position x. For all copies d, a formula $\phi_\preceq^{c,d}(x,y)$ defines the order between the c-th copy of x and the d-th copy of y. One may want to filter out some copies of the input positions. According to Courcelle's definition, this is done by having formulas $\phi_{\text{filter}}^c(x)$ which holds false if the c-th copy of x is *not* part of the output word. To shorten our definition, we assume that the c-th copy of x does not belong to the output word iff it has no label, *i.e.* none of the formula $\phi_\sigma^c(x)$ holds true.

MSO- *and* FO-*transducers* An *MSO-transducer* (resp. *FO-transducer*) over Ξ_Σ is a tuple:

$$\mathcal{T} = (k, \phi_{\text{dom}}, (\phi_\sigma^c(x))_{1 \le c \le k, \sigma \in \Sigma}, (\phi_\preceq^{c,d}(x,y))_{1 \le c,d \le k})$$

where k is an integer, ϕ_{dom}, ϕ_σ^c and $\phi_\preceq^{c,d}$ for all $c, d \in \{1, \ldots, k\}$ and $\sigma \in \Sigma$ are MSO-formulas (resp. FO-formulas) over Ξ_Σ. The MSO-transducer $\mathcal{T}$ defines a function $[\![\mathcal{T}]\!]$ of domain $[\![\phi_{\text{dom}}]\!] \subseteq \Sigma^*$, from words to structures in Ξ_Σ. For $w \in [\![\phi_{\text{dom}}]\!]$, we let the output structure $[\![\mathcal{T}]\!](w) = N = (D^N, (\sigma^N)_{\sigma \in \Sigma}, \preceq^N)$ with:

- $D^N = \{(i,c) \in \text{dom}(w) \times \{1, \ldots, k\} \mid w \models \bigvee_{\sigma \in \Sigma} \phi_\sigma^c(i)\}$
- $\sigma^N = \{(i,c) \in D^N \mid w \models \phi_\sigma^c(i)\}$, for $\sigma \in \Sigma$
- $\preceq^N = \{((i,c),(j,d)) \in D^N \times D^N \mid w \models \phi_\preceq^{c,d}(i,j)\}$

Observe that nothing guarantees that the output structure is isomorphic to a word structure. However in this paper, we assume that MSO-transducers $\mathcal{T}$ always produce structures that are isomorphic to a word structure, *i.e.*, $[\![\mathcal{T}]\!]$ is word to word transduction. This property is decidable (Filiot 2015).

Example 4. As an example, we consider the transduction $f : (ab)^n \mapsto (aab)^n$ defined by the transducer of Fig. 1(c). This transduction can be defined by an FO-transducer with only two copies of the input word. The first two a are produced while reading the first a and the b is produced while reading the b. The transduction of a particular input word by the FO-transducer we are going to construct is depicted on Fig. 6. The domain of f is $(ab)^*$, which is well-known to be FO-definable (see Example 3). Then, one defines

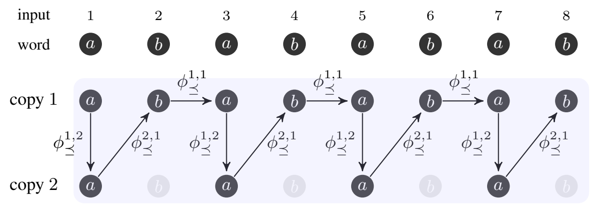


Figure 6. $\mathsf{FO}_{\mathsf{op}}$-transducer defining the transduction of Fig. 1(c)

the label formulas as follows:

$$\phi_a^1(x) \equiv \phi_a^2(x) \equiv a(x) \qquad \phi_b^1(x) \equiv b(x) \qquad \phi_b^2(x) \equiv \bot$$

Note that $\phi_b^2(x)$ is false and $\phi_a^2(x) \equiv a(x)$, and therefore the second copy of input position labeled b is not used in the output word. The order predicate is defined by:

$$\begin{array}{l} \phi_\preceq^{1,1}(x,y) \equiv \phi_\preceq^{2,2}(x,y) \equiv x \preceq y \quad \phi_\preceq^{2,1}(x,y) \equiv x \preceq y \wedge x \neq y \\ \phi_\preceq^{1,2}(x,y) \equiv (x = y \wedge a(x)) \vee (x \preceq y \wedge x \neq y) \end{array}$$

In the figure, only the successor relation of the order is depicted.

As we have said, MSO-transducers are much more expressive than rational transductions. To capture rational transductions, the following restriction can be imposed:

Order-preserving transducers An MSO-transducer (resp. FO-transducer) is called *order-preserving* (denoted by $\mathsf{MSO}_{\mathsf{op}}$ and $\mathsf{FO}_{\mathsf{op}}$ resp.) if the sentence $\forall x, y \bigwedge_{1 \le c,d \le k} (\phi_\preceq^{c,d}(x,y) \rightarrow x \preceq y)$ is valid in Σ^*. E.g. the FO-transducer defined in Example 4 and depicted in Figure 6 is an $\mathsf{FO}_{\mathsf{op}}$-transducer. The following is known:

Theorem 5. *(Bojanczyk 2014; Filiot 2015) A transduction f is rational iff it is realizable by an $\mathsf{MSO}_{\mathsf{op}}$-transducer.*

4.3 $\mathsf{FO}_{\mathsf{op}}$-transducers and aperiodicity

The proof is similar to the one from (Filiot 2015) that $\mathsf{MSO}_{\mathsf{op}}$-transducers capture exactly the rational transductions and is given in the Appendix.

Proposition 7. *A transduction is $\mathbf{A}$-rational iff it is definable by an $\mathsf{FO}_{\mathsf{op}}$-transducer.*

We are ready to prove Theorem 2 of Introduction.

Proof of Theorem 2. $\mathbf{A}$ is a decidable variety (Cho and Huynh 1991). By Theorem 1, $\mathbf{A}$-rationality of transducers is decidable, and so is FO-definability by Proposition 7. ☐

Theorem 2 has interesting consequences. Since $\mathsf{MSO}_{\mathsf{op}}$-transducers and rational transductions coincide, the $\mathsf{FO}_{\mathsf{op}}$-in-$\mathsf{MSO}_{\mathsf{op}}$ definability problem is decidable:

Corollary 2. *Given an $\mathsf{MSO}_{\mathsf{op}}$-transducer, it is decidable whether there exists an $\mathsf{FO}_{\mathsf{op}}$-transducer realizing the same transduction.*

For sequential transductions, Proposition 2 stated that aperiodicity of the transduction can be decided by testing the aperiodicity of the minimal sequential transducer from (Choffrut 2003). From Theorem 2, this also provides a way to decide $\mathsf{FO}_{\mathsf{op}}$-definability:

Corollary 3. *A sequential transduction is $\mathsf{FO}_{\mathsf{op}}$-definable if and only if it is $\mathbf{A}$-sequential.*

FO-transducers are easily seen to be closed under composition, *i.e.* for any two FO-transducers $\mathcal{T}_1, \mathcal{T}_2$, there exists an FO-transducer $\mathcal{T}$ such that $[\![\mathcal{T}]\!] = [\![\mathcal{T}_1]\!] \circ [\![\mathcal{T}_2]\!]$. The main idea is to substitute the atoms occurring in the formulas of $\mathcal{T}_1$ by the formulas of $\mathcal{T}_2$ defining these atoms. We refer the reader to (Courcelle and Engelfriet 2012) for more details. It is easily shown that the order-preserving restriction is preserved by the latter transformation whenever $\mathcal{T}_1$ and $\mathcal{T}_2$ are order-preserving. Therefore we get, as a consequence of Proposition 7, the following corollary.

Corollary 4. *Aperiodic rational transductions are closed under composition.*

Let us mention that in (Carton and Dartois 2015), it is shown that deterministic two-way aperiodic transducers are closed under composition. By inspecting the construction of (Carton and Dartois 2015), applied to aperiodic one-way transducers, one can see that this construction yields an aperiodic one-way transducer, thus proving the latter corollary, in a different way.

***Remark on other fragments of* MSO** Concerning languages, there are other results that link $\mathcal{F}$ a logical fragment of MSO to an algebraic variety $\mathbf{V}$, and the question is: does the same link hold between $\mathcal{F}_{\mathsf{op}}$-transducers and $\mathbf{V}$-transducers? Two examples are FO^2, first-order logic with only two variables, and $\mathcal{B}\Sigma_1$, first order logic with no quantifier alternation, which both have an equivalent in terms of algebraic varieties (see *e.g.* (Diekert et al. 2008)). Unfortunately, the proof of Theorem 7 (in Appendix B)linking order-preserving FO-transducers to $\mathbf{A}$-transducers cannot be simply transferred to these logics. Indeed for an FO^2 formula ϕ one cannot construct the formula $\phi_{|<x}$ without possibly adding a third variable. On the other hand, for a $\mathcal{B}\Sigma_1$ formula with a free variable, one cannot freely quantify to obtain a closed formula. For FO^2 and other logics where this problem arises, one could consider interpretations on pointed words, *i.e.* logics with an additional predicate of arity 0 (*i.e.* a constant), which is trivially equivalent to considering formulas with a free variable when the number of variables is unbounded. For $\mathcal{B}\Sigma_1$ and other logics with restricted quantification, the difficult part is to avoid additional quantification when getting rid of the constant symbol, but this problem should be discussed elsewhere.

5. Conclusion

Discussion on Theorem 1 Recall that a $\mathbf{V}$-transduction is a transduction defined by an unambiguous $\mathbf{V}$-transducer. What happens if we change the target class, say, to functional $\mathbf{V}$-transducer (not necessarily unambiguous)? It is known that any transduction can be defined by some unambiguous transducer, see for instance (Eilenberg 1974; Berstel and Boasson 1979) for a lexicographic disambiguation construction. However, the lexicographic disambiguation does not necessarily preserve the $\mathbf{V}$-membership in general. This raises the question of whether there exists a disambiguation construction that preserves varieties and more generally, whether Theorem 1 still holds if one targets functional $\mathbf{V}$-transducers instead of unambiguous $\mathbf{V}$-transducers.

FO *in* MSO *definability problem* We proved that for any rational transduction described by a transducer (thus $\mathsf{MSO}_{\mathsf{op}}$-definable), one can decide whether it is $\mathsf{FO}_{\mathsf{op}}$-definable. Hence we solve the $\mathsf{FO}_{\mathsf{op}}$ in $\mathsf{MSO}_{\mathsf{op}}$ definability problem.

A possible extension is the FO in $\mathsf{MSO}_{\mathsf{op}}$ definability problem: given a transducer, is it FO-definable? We conjecture that for rational functions, $\mathsf{FO} = \mathsf{FO}_{\mathsf{op}}$. Indeed, for deterministic two-way transducers, first-order definability and aperiodicity coincide (Carton and Dartois 2015). It remains to prove that a rational transduction given by an aperiodic two-way transducer is also definable by an aperiodic one-way transducer. This would be true if, for instance, the procedure in (Filiot et al. 2013) preserves aperiodicity.

A more involved question is the FO in MSO definability: given a two-way transducer, is its transduction FO-definable? For deterministic two-way transducers, a notion of transition monoid is defined and studied in (Carton and Dartois 2015), but no canonical object is known.

Acknowledgments

The authors warmly thank Anca Muscholl for her contribution in early stages of this work.

References

R. Alur and P. Cerný. Expressiveness of streaming string transducers. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 8 of *LIPIcs*, pages 1–12, 2010.

M. Béal and O. Carton. Determinization of transducers over finite and infinite words. *Theor. Comput. Sci.*, 289(1):225–251, 2002.

J. Berstel and L. Boasson. Transductions and context-free languages. *Ed. Teubner*, pages 1–278, 1979.

M. Bojanczyk. Transducers with origin information. In *41st Internationl Colloquium on Automata, Languages, and Programming (ICALP)*, volume 8573 of *LNCS*, pages 26–37. Springer, 2014.

M. Cadilhac, A. Krebs, M. Ludwig, and C. Paperman. A circuit complexity approach to transductions. In *Mathematical Foundations of Computer Science*, volume 9234 of *LNCS*, pages 141–153. Springer, 2015.

O. Carton and L. Dartois. Aperiodic two-way transducers and FO-transductions. In *Computer Science Logic (CSL)*, volume 41 of *LIPIcs*, pages 160–174, 2015.

S. Cho and D. T. Huynh. Finite-automaton aperiodicity is pspace-complete. *Theor. Comput. Sci.*, 88(1):99–116, 1991.

C. Choffrut. Minimizing subsequential transducers: a survey. *Theor. Comput. Sci.*, 292(1):131–143, 2003.

B. Courcelle. Monadic second-order definable graph transductions: A survey. *Theor. Comput. Sci.*, 126:53–75, 1994.

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

V. Diekert and P. Gastin. First-order definable languages. In *Logic and Automata: History and Perspectives*, volume 2 of *Texts in Logic and Games*, pages 261–306. Amsterdam University Press, 2008.

V. Diekert and M. Kufleitner. Omega-rational expressions with bounded synchronization delay. *Theory Comput. Syst.*, 56(4):686–696, 2015.

V. Diekert, P. Gastin, and M. Kufleitner. A survey on small fragments of first-order logic over finite words. *Int. J. Found. Comput. Sci.*, 19(3): 513–548, 2008.

H. Ebbinghaus and J. Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer, 1995.

S. Eilenberg. *Automata, Languages, and Machines. Volume A*. Pure and Applied Mathematics. Academic press, 1974.

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

E. Filiot. Logic-automata connections for transformations. In *Logic and Its Applications (ICLA)*, pages 30–57. Springer, 2015.

E. Filiot, O. Gauwin, P. Reynier, and F. Servais. From two-way to one-way finite state transducers. In *Logic in Computer Science (LICS)*, pages 468–477. IEEE, 2013.

E. Filiot, S. N. Krishna, and A. Trivedi. First-order definable string transformations. In *Foundation of Software Technology and Theoretical Computer Science, (FSTTCS)*, volume 29 of *LIPIcs*, pages 147–159, 2014.

Gurari and Ibarra. A note on finite-valued and finitely ambiguous transducers. *Mathematical Systems Theory*, 16, 1983.

C. Lautemann, P. McKenzie, T. Schwentick, and H. Vollmer. The descriptive complexity approach to LOGCFL. In *16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1563 of *LNCS*, pages 444–454. Springer, 1999.

L. Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

P. McKenzie, T. Schwentick, D. Thérien, and H. Vollmer. The many faces of a translation. *J. Comput. Syst. Sci.*, 72(1):163–179, 2006.

R. McNaughton and S. Papert. *Counter-free automata*. M.I.T. Press, 1971.

J. Pin and H. Straubing. Some results on $\mathcal{C}$-varieties. *ITA*, 39(1):239–262, 2005. doi: 10.1051/ita:2005014.

C. Reutenauer and M. Schützenberger. Minimization of rational word functions. *SIAM J. Comput.*, 20(4):669–685, 1991.

C. Reutenauer and M. Schützenberger. Variétés et fonctions rationnelles. *Theoretical Computer Science*, 145(1–2):229 – 240, 1995.

J. Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

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

M. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.

H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, Boston, Basel and Berlin, 1994.

A. Determinization of aperiodic transducers

Proof of Proposition 2. In order to prove Proposition 2 we use the determinization algorithm of (Béal and Carton 2002) and show that it preserves the aperiodicity of the transition congruence.

The idea of the algorithm is similar to the subset construction for automata, but taking the outputs into account: on a transition from a subset to another, the output is the longest common prefix for all the possible transitions. The rest of the outputs have to be remembered in the states themselves. Since not all rational transductions are sequential, the algorithm may not terminate. However it is shown in (Béal and Carton 2002) that if the transduction is sequential, the algorithm does terminate. Now let us describe the algorithm: Let $\mathcal{T} = (\mathcal{A}, o, i, t)$ be a transducer realizing a sequential transduction f. Let us assume that $\mathcal{A} = (Q, I, F, \Delta)$ is an $\mathbf{A}$-automaton. We give a construction of $\mathcal{T}' = (\mathcal{A}', o', i', t')$ a transducer realizing f, with $\mathcal{A}' = (Q', S_0, F', \Delta')$ being deterministic. Let $j = \wedge\{i(q) \mid q \in I\}$. Then $S_0 = \{(q, w) \mid q \in I \text{ and } i(q) = jw\}$. From the initial state we build the states and the transitions of $\mathcal{A}'$ inductively. Let S_1 be a state already constructed and let $\sigma \in \Sigma$. We define $R_2 = \{(p, vu) \mid (q, v) \in S_1 \text{ and } p \xrightarrow{\sigma|u}_{\mathcal{T}} q\}$. Let $s = \wedge\{w \mid (q, w) \in R_2\}$. Then we define a new state of Q', $S_2 = \{(q, w) \mid (q, sw) \in R_2\}$ and add the transition to Δ' and the output of the transition: $S_1 \xrightarrow{\sigma|s}_{\mathcal{T}'} S_2$. Since f is sequential, the construction must terminate, and we only have left to describe:

- $i'(S_0) = j$
- $F' = \{S \in Q' \mid \exists q \in F, (q, w) \in S\}$
- $t'(S) = wt(q)$ such that $q \in F$ and $(q, w) \in S$

The definition of t' may seem ambiguous but since f is functional it is well-defined.

Let us show that this construction preserves the aperiodicity of $\mathcal{T}$. We will show that $\mathcal{A}'$ is *counter-free i.e.* for any state S, any word u, any integer $n > 0$, if $S \xrightarrow{u^n}_{\mathcal{A}'} S$ then $S \xrightarrow{u}_{\mathcal{A}'} S$. This condition is sufficient for an automaton to be aperiodic (see *e.g.* (Diekert and Gastin 2008)). For two words u, v, let us denote by $u < v$ that u is a suffix of v and by $u \lesseqgtr v$ that either $u < v$ or $v < u$.

Let $R_0 \in Q'$, let $u \in \Sigma^+$ and let $k > 0$ such that $R_0 \xrightarrow{u^k}_{\mathcal{A}'} R_0$ and we can assume that k is the smallest of such integers. We have to show that $k = 1$. Let

$$R_0 \xrightarrow{u|x_1}_{\mathcal{T}'} R_1 \xrightarrow{u|x_2}_{\mathcal{T}'} R_2 \ldots R_{k-1} \xrightarrow{u|x_0}_{\mathcal{T}'} R_0$$

be the corresponding sequence of transitions of $\mathcal{T}'$. Since $\mathcal{A}$ is aperiodic, there is an integer n such that for any word w, $w^n \approx_{\mathcal{A}} w^{n+1}$. Let us remark that if we have $S \xrightarrow{w}_{\mathcal{A}'} R$ then the states of R are, by construction, exactly the states reachable in $\mathcal{A}$ from the a state of S. The states reachable in $\mathcal{A}$ from a state of R_0 by reading u^{kn+j} are the same for any $0 \leq j < k$, hence all the R_j's contain the same states. Let $R_j = \{(q_1, v_{1,j}), \ldots, (q_m, v_{m,j})\}$ be pairwise distinct for $0 \leq j < k$. Let us assume not all the $x_j's$ are empty otherwise the conclusion is immediate. Let $q_i \xrightarrow{u|x_{i,i'}}_{\mathcal{T}} q_{i'}$ denote, when they exist, the transitions of $\mathcal{T}$ for $1 \leq i, i' \leq m$. By construction of $\mathcal{T}'$, we have for any $0 \leq j < k$:

$$v_{i,j} x_{i,i'} = x_{j+1} v_{i',j+1}$$

Let $q_{i_0} \xrightarrow{u}_{\mathcal{A}} q_{i_1} \ldots q_{i_{t-1}} \xrightarrow{u}_{\mathcal{A}} q_{i_t}$ be a sequence of transitions in $\mathcal{A}$ such that t is a multiple of k. Then we obtain for any $0 \leq j, j' < k$:

$$v_{i_0,j} \cdot x_{i_0,i_1} \cdots x_{i_{t-1},i_t} = x_{j+1} \cdots x_{j+t} \cdot v_{i_t,j}$$
$$v_{i_0,j'} \cdot x_{i_0,i_1} \cdots x_{i_{t-1},i_t} = x_{j'+1} \cdots x_{j'+t} \cdot v_{i_t,j'}$$

In particular, the two words have a common suffix which means that $v_{i_t,j}$ and $v_{i_t,j'}$ have a common suffix. This suffix can be arbitrarily large since t can be chosen arbitrarily large. Hence $v_{i_t,j} \lesseqgtr v_{i_t,j'}$ for any such i, j, j' (any state q_t is reachable by a long enough sequence).

Let $X_0 = x_0 \cdots x_{k-1}$ and for $1 \leq j < k$ let $X_j = x_j \cdots x_{k-1} x_0 \cdots x_{j-1}$. Let us remark that $X_j x_j = x_j X_{j+1}$. Since Q is finite, for a large enough integer t, there is a state q_i, with $1 \leq i \leq m$, such that $q_i \xrightarrow{u^t}_{\mathcal{A}} q_i$. By aperiodicity, we have $q_i \xrightarrow{u^{t+1}}_{\mathcal{A}} q_i$ and we can assume that t is a multiple of k, $t = sk$. Let $i = i_1, i_2, \ldots, i_t, i$ and $i = i'_1, i'_2, \ldots, i'_{t+1}, i$ be the index sequences corresponding with these two runs. Let $Y = x_{i_1,i_2} \ldots x_{i_t,i}$ and let $Y' = x_{i'_1,i'_2} \ldots x_{i'_{t+1},i}$. Then we have for any $0 \leq j < k$:

$$\begin{array}{rcl} v_{i,j} Y & = & (X_j)^s v_{i,j} \\ v_{i,j+1} Y & = & (X_{j+1})^s v_{i,j+1} \\ v_{i,j} Y' & = & (X_j)^s x_j v_{i,j+1} \end{array}$$

We know that $v_{i,j} \lesseqgtr v_{i,j+1}$, let us first assume that $v_{i,j} < v_{i,j+1}$ and write $v_{i,j+1} = w v_{i,j+1}$. We obtain:

$$\begin{array}{rcll} v_{i,j} Y & = & (X_j)^s v_{i,j} & (\alpha) \\ w v_{i,j} Y & = & (X_{j+1})^s w v_{i,j} & (\beta) \\ v_{i,j} Y' & = & (X_j)^s x_j w v_{i,j} & (\gamma) \end{array}$$

From (α) and (β) we have: $w(X_j)^s = (X_{j+1})^s w$. By multiplying by x_j and using $X_j x_j = x_j X_{j+1}$, we have: $x_j w (X_j)^s = (X_j)^s x_j w$.

From (γ), by multiplying by Y' on the right k times we have: $v_{i,j}(Y')^k = ((X_j)^s x_j w)^k v_{i,j}$. Moreover, since $(Y')^k$ corresponds to a sequence of length a multiple of k, we also have: $v_{i,j}(Y')^k = (X_j)^{ks+1} v_{i,j}$. Hence $(X_j)^{ks+1} = ((X_j)^s x_j w)^k$. And by combining the two we obtain $(X_j)^{ks+1} = (X_j)^{ks}(x_j w)^k$, hence:

$$X_j = (x_j w)^k$$

This yields

$$\begin{aligned} (X_{j+1})^s w &= w(X_j)^s \\ &= w(x_j w)^{ks} \\ &= (w x_j)^{ks} w \end{aligned}$$

Hence

$$X_{j+1} = (w x_j)^k$$

Let q_l be another looping state state such that $q_l \xrightarrow{u^{t'}}_{\mathcal{A}} q_l$. Again, since $\mathcal{A}$ is aperiodic, by taking t' large enough we can assume, $t = t'$. If $v_{l,j} < v_{l,j+1}$ then $v_{l,j+1} = w_l v_{l,j}$. However, it also implies that $(x_j w)^k = (x_j w_l)^k$, hence $w_l = w$. On the other hand, let us assume $v_{l,j} > v_{l,j+1}$ with $w_l v_{l,j+1} = v_{l,j}$. Let Y_l and Y'_l correspond this time to sequences associated with $q_l \xrightarrow{u^t}_{\mathcal{A}} q_l$ and $q_l \xrightarrow{u^{t-1}}_{\mathcal{A}} q_l$, respectively.

$$\begin{array}{rcl} w_l v_{l,j+1} Y_l & = & (X_j)^s w_l v_{l,j+1} \\ v_{l,j+1} Y_l & = & (X_{j+1})^s v_{l,j+1} \\ v_{l,j+1} Y'_l & = & (X_{j+1})^s x_{j+1} \cdots x_{j-1} w_l v_{l,j+1} \end{array}$$

As above, the first two equations give: $(X_j)^s w_l = w_l (X_{j+1})^s$

From the third equation we have:

$$((X_{j+1})^s x_{j+1} \cdots x_{j-1} w_l)^k = (X_{j+1})^{ks+k-1}$$

Let us remark that:

$$(X_{j+1})^s x_{j+1} \cdots x_{j-1} = x_{j+1} \cdots x_{j-1} (X_j)^s$$

Hence:

$$(X_{j+1})^s x_{j+1} \cdots x_{j-1} w_l = x_{j+1} \cdots x_{j-1} w_l (X_{j+1})^s$$

From this we obtain $(x_{j+1} \cdots x_{j-1} w_l)^k = (X_{j+1})^{k-1}$.

Now from above we can write:

$$\begin{array}{rcl} (x_{j+1} \cdots x_{j-1} w_l)^k & = & ((w x_j)^k)^{k-1} \\ x_{j+1} \cdots x_{j-1} w_l & = & (w x_j)^{k-1} \\ w x_j x_{j+1} \cdots x_{j-1} w_l & = & (w x_j)^k \\ w X_j w_l & = & X_{j+1} \end{array}$$

Hence by length, we obtain $w = w_l = \varepsilon$.

Let q_l be a state reachable from q_i (any state is reachable from some such looping state). Then it can be reached by a sequence of length t and there is a corresponding sequence of outputs denoted by Z_l such that:

$$\begin{array}{rcl} v_{i,j} Z_l & = & (X_j)^s v_{l,j} \\ w v_{i,j} Z_l & = & (X_{j+1})^s v_{i,j+1} \end{array}$$

Hence $w v_{l,j} = v_{l,j+1}$.

Finally, w is a common prefix to all $v_{i',j+1}$ (j fixed and $i' \in \{1, \cdots, m\}$) and must by definition be empty. The reasoning is the same if we assume $v_{i,j} > v_{i,j+1}$. Hence $v_{i',j} = v_{i',j+1}$ and therefore $k = 1$, which means that $\mathcal{A}'$ is counter-free and thus is aperiodic. We have shown that determinization of an aperiodic transducer realizing a sequential transduction preserves the aperiodicity of the transition congruence. □

B. Equivalence between A-transducers and $\mathsf{FO_{op}}$-transducers

Proof of Proposition 7. We use the proof from (Filiot 2015) that $\mathsf{MSO_{op}}$-transducers capture exactly the rational transductions and adapt it to the aperiodic/FO case. We give the general idea and mainly focus on the parts that are relevant for the specific aperiodic case. For a more complete proof we refer the reader to the original paper.

Let $\phi(x_1, \ldots, x_k)$ be an FO formula, and let x be a variable which does not appear in ϕ. Then we define by induction $\phi(x_1, \ldots, x_k, x)_{|<x}$ a formula with one more free variable such that for any word u, for any tuple $\bar{n}$ in $\mathrm{dom}(u)^k$, for any word $v \neq \epsilon$, we have:

$$u \models \phi(\bar{n}) \Leftrightarrow uv \models \phi_{|<x}(\bar{n}, |u|+1)$$

$$\begin{array}{lcl} \phi \text{ atomic} & \rightarrow & \phi_{|<x} = \phi \\ \phi = \psi^1 \wedge \psi^2 & \rightarrow & \phi_{|<x} = \psi^1_{|<x} \wedge \psi^2_{|<x} \\ \phi = \neg\psi & \rightarrow & \phi_{|<x} = \neg\psi_{|<x} \\ \phi = \exists y\ \psi & \rightarrow & \phi_{|<x} = \exists y\ (y < x) \wedge \psi_{|<x} \end{array}$$

Let f be a transduction realized by an unambiguous aperiodic transducer $\mathcal{T} = (\mathcal{A}, o, i, t)$ with $\mathcal{A} = (Q, I, F, \Delta)$. Up to the image of the empty word, we can assume that the functions i and t are constant such that for any $q \in Q$, $i(q) = t(q) = \epsilon$. Let $q \in Q$, we define the automata $\mathcal{A}_q = (Q, \{q\}, F, \Delta)$ and $\mathcal{A}^q = (Q, I, \{q\}, \Delta)$, which are all aperiodic since they have the same transition congruence as $\mathcal{A}$. Using the Schützenberger, McNaughton-Papert theorem, we also define ϕ, ϕ_q and ϕ^q first-order formulas recognizing the same languages as $\mathcal{A}$, $\mathcal{A}_q$ and $\mathcal{A}^q$, respectively.

Let $t = (p, \sigma, q)$ be a transition of $\mathcal{A}$. We can define the formula $\psi_t(x) = \phi^p_{|<x} \wedge \sigma(x) \wedge \phi_{q|<x}$. A word u satisfies $\psi_t(i)$ if and only if there is an accepting run r of $\mathcal{A}$ on u such that $t = (r[i], u[i], r[i+1])$. From this we can define formally $\mathcal{T}' = (k, \phi_{\mathrm{dom}}, (\phi^c_\sigma(x))_{1 \leq c \leq k, \sigma \in \Sigma}, (\phi^{c,d}_{\preceq}(x,y))_{1 \leq c,d \leq k})$.

- $k = \max\{|o(\delta)| \mid \delta \in \Delta\}$
- $\phi_{\mathrm{dom}} = \phi$.

- $\phi_\sigma^c(x) = \bigvee\{\psi_t(x) \mid o(t) = v \text{ and } v(c) = \sigma\}$ for $\sigma \in \Sigma$ and $c \leq k$
- $\phi_\preceq^{c,d}(x,y) = (x = y \wedge c \leq d) \vee (x \prec y)$ for $c, d \leq k$

where $c \leq d$ is a boolean constant $\top$ or $\bot$. $\mathcal{T}'$ is indeed order-preserving and one can check that $[\![\mathcal{T}]\!] = [\![\mathcal{T}]\!]'$.

Conversely, consider the $\mathsf{FO}_{\mathsf{op}}$-transducer

$$\mathcal{T}' = (k, \phi_{\text{dom}}, (\phi_\sigma^c(x))_{1\leq c\leq k, \sigma\in\Sigma}, (\phi_\preceq^{c,d}(x,y))_{1\leq c,d\leq k})$$

realizing a transduction f. Let n be an integer and let $u = \sigma_1 \dots \sigma_n \in \Sigma^n$ be a word in the domain of f. Let $v = f(u)$ and since $\mathcal{T}$ is order-preserving, v can be decomposed as $v = v_1 \dots v_n$ such that for $|v_i| \leq k$ for any $i \leq n$. The word v_i is the sub-word of v induced by the copies of the i^{th} position of u, *i.e.* the set $N^i = \{(i,c) \mid u \models \vee_{\sigma\in\Sigma}\phi_\sigma^c(i)\}$. Let $\Sigma^{(k)}$ denote the set of words of length less than or equal to k. It means that for any word $w \in \Sigma^{(k)}$, one can define an FO formula $\psi_w(x)$ such that for any word $u \in \text{dom}(f)$ and $i \leq n$, $u \models \psi_w(i)$ if and only if $w = v_i$ according to the decomposition given above.

Now we can define an automaton $\mathcal{A}$ on the alphabet $\Gamma = \Sigma \times \Sigma^{(k)}$ which accepts all words $(\sigma_1, v_1) \dots (\sigma_n, v_n)$ such that $\sigma_1 \dots \sigma_n \models \phi_{\text{dom}}$ and for any i, v_i is the unique word such that $\sigma_1 \dots \sigma_n \models \psi_{v_i}(i)$. From this we can construct $\mathcal{T}'$ an unambiguous transducer such that for any transition $p \xrightarrow{(\sigma,v)}_{\mathcal{A}} q$ of $\mathcal{A}$, we have a transition $p \xrightarrow{\sigma|v}_{\mathcal{T}'} q$ in $\mathcal{T}'$. In order to construct an aperiodic automaton $\mathcal{A}$ we use the Schützenberger, McNaughton-Papert theorem since the language of $\mathcal{A}$ is FO-definable:

$$\phi_{\mathcal{A}} = [\phi_{\text{dom}}]_\Gamma \wedge \forall x \bigwedge_{(\sigma,w)\in\Gamma} (\sigma, w)(x) \to [\psi_w(x)]_\Gamma$$

where $[\phi]_\Gamma$ is the formula over Ξ_Γ obtained from a formula over Ξ_Σ by replacing every atomic formula $\sigma(x)$ by $\bigvee_{w\in\Sigma^{(k)}} (\sigma, w)(x)$. One can check that $[\![\mathcal{T}]\!] = [\![\mathcal{T}']\!]$.

Now we only have left to show that the automaton $\mathcal{A}'$ of $\mathcal{T}'$ is aperiodic. We will show that it is counter-free. Let $u = \sigma_1 \dots \sigma_n$ be a word of length n, let q be a state of $\mathcal{A}$ and let p be a positive integer such that $q \xrightarrow{u^p}_{\mathcal{A}'} q$. This means that for any $i \leq n$, $j \leq p$ there is a word $v_{i,j} \in \Sigma^{(k)}$ such that $q \xrightarrow{U}_{\mathcal{A}} q$ with $U = (\sigma_1, v_{1,1}) \dots (\sigma_n, v_{n,1})(\sigma_1, v_{1,2}) \dots (\sigma_n, v_{n,p})$. Without loss of generality we can assume that q is both accessible and co-accessible. Let w_1, w_2 be words such that $q_0 \xrightarrow{w_1}_{\mathcal{A}} q$ and $q \xrightarrow{w_2}_{\mathcal{A}} q_F$ where q_0 and q_F respectively are an initial state and a final state of $\mathcal{A}$. Since the formulas $\psi_w(x)$ are FO, we can show using Ehrenfeucht-Fraïssé games (see *e.g.* (Libkin 2004)) that there exists an integer m such that for any words x, y, z, (y being non-empty) and for $i \leq |y|$ we have $xy^m yy^m z \models \psi(|x| + |y|^m + i) \Rightarrow xy^m yy^m z \models \psi(|x| + |y|^{m+1} + i)$. Since we have for any $i \leq n$, $j \leq p$, $w_1 u^{km} u^k u^{km} w_2 \models \psi_{v_{i,j}}(|w_1| + |u|^{km+(j-1)} + i)$, we get that $w_1 u^{km} u^k u^{km} w_2 \models \psi_{v_{i,j}}(|w_1| + |u|^{km+j} + i)$. This means that we have $v_{i,j} = v_{i,j+1}$ for any $1 \leq j < p$, so let $v_i = v_{i,j}$. Hence, $q \xrightarrow{((\sigma_1,v_1)\dots(\sigma_n,v_n))^p}_{\mathcal{A}} q$. However, $\mathcal{A}$ is a deterministic and aperiodic automaton and is thus counter-free, hence $q \xrightarrow{(\sigma_1,v_1)\dots(\sigma_n,v_n)}_{\mathcal{A}} q$. Finally we obtain that $q \xrightarrow{u}_{\mathcal{A}'} q$ which means that $\mathcal{A}'$ is counter-free and $\mathcal{T}'$ is an unambiguous $\mathbf{V}$-transducer. $\square$

C. Closure of $\mathsf{FO}_{\mathsf{op}}$-transducers under composition

Proof of Corollary 4. We want to show that $\mathsf{FO}_{\mathsf{op}}$-transducers are closed under composition. It is well-known that MSO-transducers, as well as FO-transducers, are closed under composition (Courcelle and Engelfriet 2012). Let us give the overall idea. If the set of copies of $\mathcal{T}_1$ is $\{1, \dots, k_1\}$ and the one of $\mathcal{T}_2$ is $\{1, \dots, k_2\}$, then the set of copies of $\mathcal{T}$, which realizes the composition, is $\{1, \dots, k_1\} \times \{1, \dots, k_2\}$, which is isomorphic to $\{1, \dots, k_1k_2\}$. Then, an output position of the image of a word w by $\mathcal{T}$ is a triple (i, c_1, c_2) where $i \in \text{dom}(w)$, c_1 is a copy of $\mathcal{T}_1$ and c_2 is a copy of $\mathcal{T}_2$. Then, to give a very high-level idea, the formulas of $\mathcal{T}$ are obtained by substituting atomic propositions in the formulas of $\mathcal{T}_1$ by formulas of $\mathcal{T}_2$, and by restraining the quantifiers of the formulas of $\mathcal{T}_1$ to positions that exist by $\mathcal{T}_2$. We refer the reader to (Courcelle and Engelfriet 2012) for all the details, but extract the main argument that will allow us to show that the order-preserving restriction is preserved: the construction of (Courcelle and Engelfriet 2012) preserves *origin information*, in the sense of (Bojanczyk 2014).

Given $(w, w') \in [\![\mathcal{T}_1]\!]$, the origin mapping is a function o from $\text{dom}(w')$ to $\text{dom}(w)$, defined as follows: every position in w' is a pair (i, c) where $i \in \text{dom}(w)$ and c is a copy of $\mathcal{T}_1$. Then, the origin of node (i, c) is defined as i, *i.e.* $o(i, c) = i$. Moreover, $\mathcal{T}_1$ is order-preserving iff o is order-preserving, in the sense that for any two elements α, β in the range of o, if $\alpha \preceq \beta$, then $o(\alpha) \preceq o(\beta)$.

Now, it is easily seen that the construction of (Courcelle and Engelfriet 2012) preserves origin, in the following sense: if $(w_1, w_3) \in [\![\mathcal{T}]\!]$ with origin mapping o, then there exists w_2 such that $(w_1, w_2) \in [\![\mathcal{T}_2]\!]$ with origin mapping o_2 and $(w_2, w_3) \in [\![\mathcal{T}_1]\!]$ with origin mapping o_1, such that $o = o_2 \circ o_1$. Finally, since o_1 and o_2 are order-preserving, so is o: Indeed, take two elements $\alpha \preceq \beta$ in the range of o, since o_1 is order-preserving, $o_1(\alpha) \preceq o_1(\beta)$ and since o_2 is order-preserving, $o_2(o_1(\alpha)) \preceq o_2(o_1(\beta))$, *i.e.* $o(\alpha) \preceq o(\beta)$. $\square$