Determinization of transducers over infinite words: the general case

Marie-Pierre Béal
Institut Gaspard Monge,
Université de Marne-la-Vallée
http://www-igm.univ-mlv.fr/~beal/

Olivier Carton
LIAFA,
Université Paris 7
http://www.liafa.jussieu.fr/~carton/

June 19, 2003

Abstract

We consider transducers over infinite words with a Büchi or a Muller acceptance condition. We give characterizations of functions that can be realized by Büchi and Muller sequential transducers. We describe an algorithm to determinize transducers defining functions over infinite words.

1 Introduction

The aim of this paper is the study of the determinization of transducers over infinite words, that is of machines realizing rational transductions over infinite words. Transducers are finite state automata with edges labeled by pairs of finite words (an input and an output label). They are very useful in a lot of areas like coding [10], computer arithmetic [11], language processing (see for instance [16] and [13]) or in program analysis [8]. Transducers that have a deterministic input automaton are called sequential transducers [19] and functional relations that can be realized by a sequential transducer are called sequential functions. They play an important role since they allow sequential encoding. The determinization of a transducer is the construction of a sequential transducer which defines the same function. We refer the reader to [4] and [18] for complete introductions to transducers.

The determinization of an automaton over finite words is easily solved by a subset construction. The determinization of a transducer is more complex than the determinization of an automaton since it involves both the input and the output labels. In the case of finite words, it has been solved by Choffrut who gives in [6, 7] a characterization of subsequential functions and an algorithm that transforms a transducer which realizes a subsequential function into a subsequential transducer (see also [4, p. 109–110], [16, p. 223–233] and [2]). Choffrut proved that the subsequentiality of functions realized by transducers over finite words is decidable. A polynomial time decision procedure has been obtained by Weber and Klemm in [22], see also [3]. The determinization of transducers over finite words is the first step before a minimization process introduced by Choffrut in [6] and [7]. Efficient algorithms to minimize sequential transducers have been described later in [13], [14], [5] and [1].

We consider here transducers that define functional relations over infinite words. The determinization of automata over infinite words is already much more difficult than over finite words. First, not every Büchi automaton can be determinized. Muller automata which have a more powerful acceptance condition must be used [12]. Second, all determinization algorithms of automata over infinite words that have been given so far are complex [17]. In [2], we have coped with this difficulty by considering transducers without acceptance condition, that is, all their states are final. This case is indeed much simpler because the determinization of an automaton over infinite words without any acceptance condition can be achieved by a simple subset construction. However, in this case, the determinization of a transducer is already non-trivial and needs new techniques like the notion of constant states. In [2], we have given a characterization of sequential functions, and a determinization algorithm, in the case where all states of the transducer are final.

In this paper, we solve the general case, that is, where the transducers over infinite words have Büchi or Muller acceptance conditions. We give characterizations of functions that can be realized by Büchi or Muller sequential transducers. In the case where the function is Büchi or Muller sequential, we give an effective algorithm to construct a sequential transducer (*i.e.*, a transducer with a deterministic input automaton) which realizes the same function. However, this result does not completely cover those in [2]. Indeed, this general algorithm applied to a transducer without acceptance condition yields a sequential transducer with an acceptance condition although we have proved in [2] that a sequential function realized by a non-sequential transducer without acceptance condition can actually be realized by a sequential transducer without acceptance condition.

The paper uses notions already considered in [2] like the notion of a constant state in a transducer (a state such that all paths going out of it have the same infinite output label) but it also introduces new methods. The characterizations are based on the continuity of the function realized by the transducer and on a new notion which is a variant of the twinning property introduced by Choffrut [6, 7] that we call weak twinning property. The determinization algorithm is performed in two main steps. The first step constructs a sequential transducer without acceptance condition which realizes an extension of the function f realized by the initial transducer. The second step combines, with an easy product construction, the transducer obtained at the first step with a deterministic Büchi (or Muller) automaton recognizing the domain of f to get a Büchi (or Muller) sequential transducer that realizes exactly f. The problem that the determinization of transducers includes the determinization of automata is thus avoided by this second step. Roughly speaking, the first step mainly deals with the outputs of the transducer whereas the second one ignores completely the outputs and deals only with the inputs.

We mention that the continuity of functions realized by Büchi transducers is decidable in polynomial time [15]. The decidability of the weak twinning property that we introduce is not discussed in the paper. See the Conclusion for a further discussion.

A consequence of our characterizations is that any function realized by a Muller sequential transducer is the restriction of a function realized by a Büchi sequential transducer. This means that the difference between functions realized by Büchi and Muller sequential transducers is entirely due to the domains of the functions and not to the outputs.

The paper is organized as follows. Basic notions about transducers and acceptance conditions over infinite words are defined in Section 2. The two main results (Theorem 3 and Theorem 4) that state the characterizations of Büchi and Muller sequential functions are given in Section 3. Section 4 contains the determinization algorithm and an example of the construction of a sequential transducer.

2 Transducers

In the sequel, A and B denote finite alphabets. The set of finite and infinite words over A are denoted by A^* and A^ω, respectively. The empty word is denoted by ε.

A *transducer* over $A \times B$ is composed of a finite set Q of *states*, a finite set $E \subset Q \times A^* \times B^* \times Q$ of *edges*, a set $I \subset Q$ of *initial* states and an acceptance condition Φ. An edge $e = (p, u, v, q)$ from p to q is denoted by $p \xrightarrow{u|v} q$. The words u and v are called the *input label* and the *output label* of the edge. Thus, a transducer is the same object as an automaton, except that the labels of the edges are pairs of words instead of letters (as usual) or words.

A *finite path* γ in a transducer is a finite sequence

$$q_0 \xrightarrow{u_1|v_1} q_1 \xrightarrow{u_2|v_2} \cdots \xrightarrow{u_n|v_n} q_n$$

of consecutive edges. Its *input label* is the word $u = u_1u_2\ldots u_n$, its *output label* is the word $v = v_1v_2\ldots u_n$ and its *label* is the pair (u, v) (also denoted $u|v$) of finite words. Such a path is sometimes denoted by $q_0 \xrightarrow{u|v} q_n$ like a transition. We say it *starts* at q_0 and *ends* at q_n. Similarly, an *infinite path* γ in a transducer is an infinite sequence

$$q_0 \xrightarrow{u_0|v_0} q_1 \xrightarrow{u_1|v_1} q_2 \xrightarrow{u_2|v_2} q_3 \cdots$$

of consecutive edges. Its *input label* is the word $x = u_0u_1u_2\ldots$, its *output label* is the word $y = v_0v_1v_2\ldots$ and its *label* is the pair (x, y) (also denoted $x|y$) of words. Note that the input label or the output label of an infinite path may be a finite word because the input label or the output label of a transition may be the empty word. We say that the path *starts* at q_0. We denote by $\lim(\gamma)$ the set of states that appear infinitely often along γ. Since the number of states of the transducer is finite, $\lim(\gamma)$ is always nonempty.

The acceptance condition Φ determines a family of final paths as follows. A path is *final* if it satisfies Φ and if both its input and output labels are infinite words. A path is *successful* if it is final and if it starts at an initial state. In this paper, we consider two types of acceptance condition : Büchi and Muller acceptance conditions. In a *Büchi transducer* the acceptance condition Φ is a set F of states, called final states, and a path γ satisfies Φ if it goes infinitely often through a final state, *i.e.*, $\lim(\gamma) \cap F \neq \varnothing$. In a *Muller transducer* the acceptance condition Φ is a family $\mathcal{F}$ of sets of states, and a path γ satisfies Φ if $\lim(\gamma) \in \mathcal{F}$. Observe that whether or not γ satisfies Φ depends only on the set $\lim(\gamma)$ of states that occur infinitely often along the path γ. Therefore, removing a finite prefix of a final path or prefixing a final path with a finite path always yields a final path.

In the sequel we say that a finite cycling path around a state q (i.e., starting and ending at q), also called a loop, is *accepting* if the infinite path made by looping infinitely often along this loop is final. For a Büchi acceptance condition, a loop is accepting if it contains a final state. For a Muller acceptance condition, a loop is accepting if the set of states that are encountered along the path belongs to the family $\mathcal{F}$.

A pair (x, y) of infinite words is *recognized* if it is the label of a successful path. The set of all recognized pairs is the relation *realized* by the transducer. This relation R is of course a function f if for any word $x \in A^\omega$, there exists at most one word $y \in B^\omega$ such that $(x, y) \in R$. In that case, a transducer can be seen as a machine computing nondeterministically the output word $y = f(x)$ from the input word x. We denote by $\mathrm{dom}(f)$ the domain of the function f.

As in the case of automata, nondeterministic Büchi and Muller transducers have the same power. First, any Büchi transducer with a set F of final states can be viewed as Muller transducer whose acceptance condition is given by the family $\mathcal{F} = \{P \subseteq Q \mid P \cap F \neq \varnothing\}$. Conversely, any Muller transducer can be simulated by a Büchi transducer. This equivalent Büchi transducer can be obtained by the same construction as for automata [21, p. 417].

A transducer is *trim* if each state is accessible from an initial state and if there is at least one final path starting at each state. States which do not satisfy these conditions can be removed. Therefore, we assume in the sequel that all transducers are trim. Note that it can be effectively checked whether a given state is accessible from an initial state. It can also be effectively checked whether it is the first state of a final path. Indeed a state is the first state of a final path if an accepting loop is accessible from that state. Therefore, a transducer can be effectively made trim. This action can be seen as a preprocessing of the transducer.

A transducer is said to be *real-time* if it is labeled in $A \times B^*$, that is, the input label of each transition is a letter. We say that a transducer $\mathcal{T}$ is *sequential* if the following conditions are satisfied:

- it is real-time,
- it has a unique initial state,
- for any state q and any letter a, there is at most one transition going out of q and input labeled by a.

These conditions ensure that for each word $x \in A^\omega$, there is at most one word $y \in B^\omega$ such that (x, y) is recognized by $\mathcal{T}$. Thus, the relation realized by $\mathcal{T}$ is a function from A^ω into B^ω. A function is said to be *Büchi sequential* (respectively *Muller sequential*) if it can be realized by a sequential Büchi (respectively Muller) transducer.

In the case of finite words, one often distinguishes sequential and subsequential functions. In a subsequential transducer, an additional finite word depending on the ending state is appended to the output label of the path. However, the notion of subsequential transducer is irrelevant in the case of infinite words.

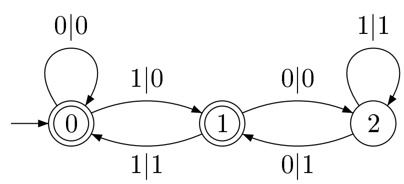
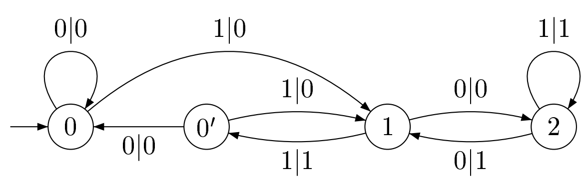


Figure 1: Sequential Büchi transducer of Example 1



$\mathcal{F} = \{\{0\}, \{0, 0', 1\}, \{0', 1, 2\}, \{0, 0', 1, 2\}, \{1, 2\}\}$

Figure 2: Sequential Muller transducer of Example 1

Example 1 Let $A = \{0, 1\}$ be the binary alphabet. Consider the sequential transducer $\mathcal{T}$ pictured in Figure 1. If the infinite word x is the binary expansion of a real number $\alpha \in [0, 1)$, the output corresponding to x in $\mathcal{T}$ is the binary expansion of $\alpha/3$. If all states of this transducer are final, it accepts both as input and as output label binary expansions which are not normalized, that is of the form $(0 + 1)^*1^\omega$. In order to reject these expansions as output label, this transducer must be equipped with the Büchi acceptance condition $F = \{0, 1\}$ as shown in Figure 1. In order to reject these expansions also as input label, the state 0 must be split and the transducer must be equipped with a Muller acceptance condition as shown in Figure 2.

The following proposition allows us in the sequel to only consider real-time transducers. This result is due to Gire [9] in the more general case of rational relations of infinite words. We give below a simpler proof for rational functions.

Proposition 2 *For any Büchi transducer realizing a function of infinite words, one can compute a real-time Büchi transducer realizing the same function.*

Proof Let $\mathcal{T}$ be a Büchi transducer realizing a function. We can assume that each transition is labeled by a pair $\varepsilon|a$ or $a|\varepsilon$ where a is a letter or ε. Otherwise, each transition $p \xrightarrow{u|v} q$ where $u = a_1 \dots a_m$ and $v = b_1 \dots b_m$ can be replaced by $n + m$ consecutive transitions

$$p \xrightarrow{a_1|\varepsilon} q_1 \xrightarrow{a_2|\varepsilon} q_2 \cdots q_{n-1} \xrightarrow{a_n|\varepsilon} q_n \xrightarrow{\varepsilon|b_1} q_{n+1} \cdots q_{m+n-1} \xrightarrow{\varepsilon|b_m} q,$$

where $q_1, \dots, q_{m+n-1}$ are new states.

Let Q be the set of states of $\mathcal{T}$ and let F be its set of final states. We define a real-time transducer $\mathcal{T}'$ as follows.

Let a be a letter of the input alphabet, let p and q be two states of $\mathcal{T}$, and let e be 0 or 1. If $e = 0$, let $V^{a,e}_{p,q}$ be the set of words v such that there is a path $p \xrightarrow{a|v} q$ from p to q with input label a and output label v. If $e = 1$, let $V^{a,e}_{p,q}$ be the set of words v such that there is a path $p \xrightarrow{a|v} q$ from p to q with input label a and output label v and which goes through a final state. Note that $V^{a,e}_{p,q}$ is always a rational subset of B^* and that $V^{a,1}_{p,q}$ is a subset of $V^{a,0}_{p,q}$. Suppose that two nonempty words v and v' belong to a set $V^{a,e}_{p,q}$. Whenever the path $p \xrightarrow{a|v} q$ occurs in a successful path of $\mathcal{T}$, it can be replaced by the path $p \xrightarrow{a|v'} q$. Indeed, since the transducer $\mathcal{T}$ realizes a function, the output word of the successful path remains unchanged. This means that it suffices to keep one nonempty word in each set $V^{a,e}_{p,q}$. From $V^{a,e}_{p,q}$, we pick a subset $W^{a,e}_{p,q}$ of cardinality at most 2 as follows.

- If $V^{a,e}_{p,q}$ contains the empty word, the empty word is also put in $W^{a,e}_{p,q}$.
- If $V^{a,e}_{p,q}$ contains at least one nonempty word, one of them is put in $W^{a,e}_{p,q}$.

The set of states of $\mathcal{T}'$ is the set $Q' = Q \times \{0, 1\}$. The set of initial states is $I' = \{(q, 0) \mid q \in I\}$ and the set of final states is $F' = \{(q, 1) \mid q \in Q\}$. The set of transitions of $\mathcal{T}'$ is defined as follows. Let a be a letter of the input alphabet and let (q, e) and (q', e') be two states of $\mathcal{T}'$. There is a transition from (q, e) to (q', e') labeled by $a|v$ if $v \in W^{a,e'}_{p,q}$. The transducer $\mathcal{T}'$ realizes the same function as $\mathcal{T}$. This is independent of the choice of the finite subsets $W^{a,e}_{p,q}$. $\square$

The domain of a function realized by a Büchi or Muller transducer is a rational set of infinite words. Recall that a set of infinite words is said to be *rational* if it is accepted by an automaton. An automaton is a transducer where the edges are labeled by letters instead of pairs of words. The label of a path in an automaton is thus a word. A Büchi (respectively Muller) automaton is an automaton equipped with a Büchi (respectively Muller) acceptance condition. We refer the reader to [20] or [21] for a complete introduction to automata on infinite words.

It is not true that any rational set of infinite words is recognized by a deterministic Büchi automaton. However, any rational set of infinite words is recognized by a deterministic Muller automaton [21, Thm 5.1]. Furthermore an equivalent deterministic Muller automaton can be computed from a Büchi automaton. Sets of infinite words that can be recognized by a deterministic Büchi automaton are called *deterministic*. It can be effectively checked whether the set of words recognized by a given Büchi automaton is deterministic [20, Thm 5.3c]. Furthermore, if that set is deterministic, an equivalent deterministic Büchi automaton can effectively be computed [20, Lem 5.4].

A Büchi automaton recognizing the domain of a function can be effectively computed from a transducer realizing the function. The rough idea is to remove the output labels of the edges. We refer the reader to the proof of the main result in [2].

3 Characterization of sequential functions

The characterizations of Büchi and Muller sequential functions need the notion of continuity of a function. First recall that the set A^ω is endowed with the usual topology. This topology can be defined by the distance d given by

$$d(x,y) = \begin{cases} 0 & \text{if } x = y \\ 2^{-n} & \text{where } n = \min\{k \mid x_k \neq y_k\} \text{ otherwise.} \end{cases}$$

Intuitively two infinite words are close if they share a long common prefix. Therefore, a sequence of infinite words $(x_n)_{n\geq 0}$ converges to a word x if for any integer k, there is an integer n_k such that any word x_n for $n \geq n_k$ has a common prefix with x of length greater than k. We recall now a definition of the continuity that we use later. A function f is said to be *continuous* if for any sequence $(x_n)_{n\geq 0}$ of elements of its domain converging to an element x of its domain, the sequence $(f(x_n))_{n\geq 0}$ converges to $f(x)$.

The characterizations of Büchi and Muller sequential functions also need the notion of a constant state in a transducer. We say that a state q of a transducer is *constant* if all final paths starting at this state have the same output label. The terminology comes from the fact that the transducer in which q is initial realizes a constant function. For a constant state q, the common output label of all final paths starting at q is denoted by y_q. This infinite word always exists since the transducer is assumed to be trim.

In order to illuminate the notion of a constant state, we make some remarks and we prove some easy properties. Note first that in the definition of a constant state, we only consider final paths. There may be other, nonfinal, infinite paths with either a finite output label or an infinite output label which is different from the output of a final path.

Note that if q is a constant state and if the state q' is accessible from q, q' is also a constant state. Indeed, suppose that there is a finite path γ from q to q' whose output label is v. If γ_1 and γ_2 are two final paths starting at q', the two paths $\gamma\gamma_1$ and $\gamma\gamma_2$ are two final paths starting at q. It follows that the output labels of γ_1 and γ_2 must be equal and q' is a constant state. Furthermore, the output labels y_q and $y_{q'}$ satisfy $y_q = vy_{q'}$.

Note also that the common output label y_q of a constant state q is an ultimately periodic word, that is an infinite word of the form uv^ω for two finite words u and v. If there is a final path starting at q, then there is always an ultimately periodic final path starting at q since the number of states is finite.

Note finally that if q is a constant state and there is a finite path from q to q (a loop) with a nonempty output label v, then the output label y_q is equal to v^ω. This is true even if the loop around q is not accepting. Let γ be the finite path from q to q with the output label v and let γ_1 be a final path starting at q. By definition, the output label of γ_1 is y_q. Since the path $\gamma\gamma_1$ is also a final path starting at q, the equality $vy_q = y_q$ holds. Since v is nonempty, y_q is equal to v^ω.

The characterization of sequentiality is essentially based on the following notion which is a variant of the twinning property introduced by Choffrut [7, p. 133] (see also [4, p. 128]). This property is a kind of compatibility of the outputs of paths with the same inputs. A transducer has the *weak twinning property* if for any pair of paths

$$\begin{aligned} i &\xrightarrow{t|u} q \xrightarrow{v|w} q \\ i' &\xrightarrow{t|u'} q' \xrightarrow{v|w'} q', \end{aligned}$$

where i and i' are initial states, the following two properties hold.

- If both q and q' are not constant, then either $w = w' = \varepsilon$ or there exists a finite word s such that either $u' = us$ and $sw' = ws$, or $u = u's$ and $sw = w's$. The latter case is equivalent to the following two conditions:

  (i) $|w| = |w'|$,
  (ii) $uw^\omega = u'w'^\omega$

- If q is not constant, q' is constant, and w is nonempty, then the equality $u'y_{q'} = uw^\omega$ holds. Note that if w' is nonempty, then $y_{q'} = w'^\omega$.

No property is required when both q and q' are constant states. In that case, the compatibility of the outputs is already ensured by the functionality of the transducer. The property required when both q and q' are not constant is exactly the twinning property as defined by Choffrut [7] (required for all q and q'). The weak twinning property and the twinning property only differ in the way constant states are treated.

We now state the two characterizations of Büchi and Muller sequential functions.

THEOREM 3 *Let f be a function realized by a real-time Büchi transducer $\mathcal{T}$. Then the function f is Muller sequential iff the following two properties hold:*

- *the function f is continuous,*
- *the transducer $\mathcal{T}$ has the weak twinning property.*

THEOREM 4 *Let f be a function realized by a real-time Büchi transducer $\mathcal{T}$. Then the function f is Büchi sequential iff the following two properties hold:*

- *the domain of f can be recognized by a deterministic Büchi automaton,*
- *the function f is Muller sequential.*

Before proceeding to the proofs of the theorems we provide some examples showing that the conditions are independent.

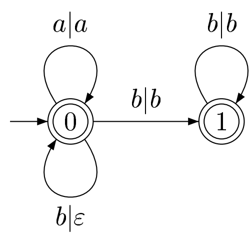


Figure 3: Transducer of Example 5

EXAMPLE 5 The Büchi transducer pictured in Figure 3 is equipped with a Büchi acceptance condition. It realizes a noncontinuous function f. Indeed, the image of an infinite word x is $f(x) = a^\omega$ if x has infinitely many occurrences of a and it is $f(x) = a^n b^\omega$ if x has n occurrences of a. Although the sequence $x_n = b^n a b^\omega$ converges to $x = b^\omega$, the sequence $f(x_n) = ab^\omega$ does not converge to $f(x) = b^\omega$. State 1 is constant but state 0 is not. This transducer has the weak twinning property. Note also that it does not have the twinning property since there are paths $0 \xrightarrow{b|\varepsilon} 0 \xrightarrow{b|\varepsilon} 0$ and $0 \xrightarrow{b|b} 1 \xrightarrow{b|b} 1$. This shows that the weak twinning property is really weaker.

EXAMPLE 6 The Büchi transducer pictured in Figure 4 realizes the continuous function defined by $f(a^\omega) = a^\omega$, $f(a^n bx) = a^n bx$ and $f(a^n cx) = a^{2n} cx$ for any $n \geq 0$ and $x \in \{a, b, c\}^\omega$. However, this transducer does not have the weak twinning property. The states 1 and 2 are not constant but one has the following paths $0 \xrightarrow{a|a} 1 \xrightarrow{a|a} 1$ and $0 \xrightarrow{a|aa} 2 \xrightarrow{a|aa} 2$.

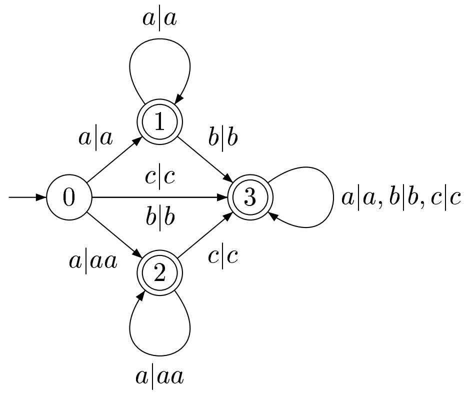


Figure 4: Transducer of Example 6

EXAMPLE 7 Let A be the alphabet $\{a, b\}$ and let $X = A^*b^\omega$ be the set of infinite words having finitely many a. Let f be the identity function restricted to the set X. This function is Muller sequential but it is not Büchi sequential since its domain is not deterministic.

The proofs of Theorems 3 and 4 are given in the remainder of the paper. We prove below that the conditions in Theorems 3 and 4 are necessary. The converse follows from the algorithm that we describe in the following section.

We first prove that a function f realized by a Muller sequential transducer $\mathcal{S}$ must be continuous. Suppose that the sequence $(x_n)_{n\geq 0}$ of infinite words converges to x and that all x_n and x are in the domain of f. Since $\mathcal{S}$ is sequential, each word of the domain is the input label of exactly one path. Let γ_n be the path labeled by x_n and let γ be the path labeled by x. Since x_n converges to x, the common prefix of x_n and x becomes longer and longer. It follows that γ_n converges to γ and hence $f(x_n)$ converges to $f(x)$.

It is almost straightforward that the domain of a Büchi sequential function f is recognized by a deterministic Büchi automaton. An infinite word belongs to the domain of f if it is the input label of a path which goes infinitely often through a final state and through a transition with a nonempty output label. A Büchi automaton recognizing the domain can be easily constructed from a sequential Büchi transducer realizing f.

It remains to prove that a transducer $\mathcal{T}$ realizing a Muller sequential function has the weak twinning property. We suppose that we have the following paths in $\mathcal{T}$.

$$\begin{aligned} i &\xrightarrow{t|u} q \xrightarrow{v|w} q \\ i' &\xrightarrow{t|u'} q' \xrightarrow{v|w'} q', \end{aligned}$$

where i and i' are initial states. Let $\mathcal{S}$ be a sequential Muller transducer realizing the same function f as $\mathcal{T}$. Let $x|y$ be the label of a final path in $\mathcal{T}$ starting at q.

For any integer n, the equality $f(tv^n x) = uw^n y$ holds. Since $\mathcal{S}$ realizes f, there must be a successful path in $\mathcal{S}$ with label $tv^n x|uw^n y$ for any n. For n greater than the number of states of $\mathcal{S}$, the same state appears twice. Then there is in $\mathcal{S}$ a path

$$i'' \xrightarrow{tv^l|u''} q'' \xrightarrow{v^k|w''} q''$$

where $l \geq 0$, $k \geq 1$, and i'' is the initial state of $\mathcal{S}$. By prolonging the path in $\mathcal{T}$ from i to q (respectively from i' to q') with l iterations of the path around q (respectively around q'), we can assume without loss of generality that $l = 0$. By replacing the cycling path around q (respectively around q') by k iterations of this path, we can also assume without loss of generality that $k = 1$.

We claim that if the state q is not constant, then the equality $|w| = |w''|$ holds. Indeed, let $x|y$ and $x'|y'$ be the labels of two final paths starting at q such that $y \neq y'$. There are in $\mathcal{S}$ two paths labeled by $x|z$ and $x'|z'$ starting at the state q'' such that for any $n \geq 0$

$$\begin{aligned} f(tv^n x) &= uw^n y \ = u''w''^n z \\ f(tv^n x') &= uw^n y' = u''w''^n z'. \end{aligned}$$

If $|w| < |w''|$, the words y and y' have a common prefix of length $|u''| - |u| + n(|w''| - |w|)$ for any large n. This leads to the contradiction that $y = y'$. If $|w''| < |w|$, the words z and z' have a common prefix of length $|u| - |u''| + n(|w| - |w''|)$ for any large n. This leads to the contradiction that $z = z'$ and $y = y'$. This proves that $|w| = |w''|$ and, if they are nonempty, that $uw^\omega = u''w''^\omega$.

We first suppose that q' is also not constant. By symmetry one has $|w| = |w''| = |w'|$. Furthermore, if they are nonempty, one has $uw^\omega = u''w''^\omega = u'w'^\omega$.

We now suppose that q' is constant and that w is nonempty. This last assumption implies that w'' is also nonempty and that the equality $uw^\omega = u''w''^\omega$ holds. Let $x'|y_{q'}$ be the label of a final path starting at q'. Then there is a path in $\mathcal{S}$ with label $x'|z''$ starting at q'' such that

$$f(tv^n x') = u'w'^n y_{q'} = u''w''^n z''$$

for any $n \geq 0$. Since q' is constant, the word $w'^n y_{q'}$ is equal to $y_{q'}$. Therefore, the word $u'y_{q'}$ is equal to $u''w''^n z''$ for any integer n. Thus, it is equal to $u''w''^\omega$ since w'' is nonempty. This ends the proof of the necessity of the conditions in Theorems 3 and 4.

4 Determinization algorithm

In this section, we describe an algorithm to determinize a Büchi transducer which satisfies the conditions of Theorem 3 or 4. We describe the construction of a sequential transducer $\mathcal{S}$ from a Büchi transducer $\mathcal{T}$. The transducer $\mathcal{S}$ has a trivial acceptance condition. This means that any infinite path in $\mathcal{S}$ which has infinite input and output labels is final. If the transducer $\mathcal{T}$ satisfies the conditions of Theorem 3, the function realized by $\mathcal{S}$ is an extension of the function realized by $\mathcal{T}$. Then it suffices to combine the transducer $\mathcal{S}$ with a Muller automaton recognizing the domain of $\mathcal{T}$ to obtain a Muller sequential transducer which realizes the same function as $\mathcal{T}$. If furthermore the domain of $\mathcal{T}$ is recognized by a deterministic Büchi automaton $\mathcal{A}$, the transducer $\mathcal{S}$ is combined with $\mathcal{A}$ to obtain a Büchi sequential transducer which realizes the same function as $\mathcal{T}$.

The sequential transducer $\mathcal{S}$ is obtained from $\mathcal{T}$ by performing a kind of subset construction. For a fixed finite word u, all states which can be accessed from the initial states by some path whose input label is u, are grouped together into a state of $\mathcal{S}$. To each of these states is associated a word. This word gives what remains to be output. For a nonconstant state, this word is finite and it is the suffix of the output obtained by deleting to the left the maximal common prefix of the outputs labelling these paths. For a constant state, this word is infinite and it equals vw where v is as in the previous case and w is the unique ultimately periodic output the state can produce. The construction yields potentially infinitely many composite states consisting of pairs (state, output word). It just happens that under the assumptions of Theorem 3 it leads to a finite object.

We now describe the sequential transducer $\mathcal{S}$. By Proposition 2, we can suppose that the transducer $\mathcal{T}$ is real-time. This means that the labels of the edges belong to $A \times B^*$. The construction can actually be adapted to deal with transducers with edges labeled by $A^* \times B^*$ but this is a bit technical. Let us denote by Q, E, I, and C the set of states, edges, initial states, and constant states of $\mathcal{T}$ respectively. A state of $\mathcal{S}$ is a finite set P containing two kinds of pairs. The first kind are pairs (q, z) where q belongs to $Q \setminus C$ and z is a finite word over B. The second kind are pairs (q, z) where q belongs to C and z is an ultimately periodic infinite word over B. We now describe the transitions of $\mathcal{S}$. Let P be a state of $\mathcal{S}$ and let a be a letter in A. Let R be equal to the set defined as follows

$$\begin{aligned} R = \{(q', zv') \mid q' \notin C \text{ and } \exists (q, z) \in P,\ q \notin C \text{ and } q \xrightarrow{a|v'} q' \in E\} \\ \cup \{(q', zv'y_{q'}) \mid q' \in C \text{ and } \exists (q, z) \in P,\ q \notin C \text{ and } q \xrightarrow{a|v'} q' \in E\} \\ \cup \{(q', z) \mid q' \in C \text{ and } \exists (q, z) \in P,\ q \in C \text{ and } q \xrightarrow{a|v'} q' \in E\}. \end{aligned}$$

There are only three cases in the definition of R because q' is constant if q is already constant. We now define the transition from the state P with input label a. If R is empty, there is no transition from P with input label a. Otherwise, the output label of this transition is the word v defined as follows. We define v as the first letter of the word z if R only contains pairs (q', z) with $q' \in C$ and all the infinite words z are equal. Otherwise, we define v as the longest common prefix of all the finite or infinite words z for $(q', z) \in R$. The state P' is defined as follows

$$P' = \{(q', z') \mid (q', vz') \in R\}.$$

Then there is a transition $P \xrightarrow{a|v} P'$ in $\mathcal{S}$. The initial state of $\mathcal{S}$ is the set J where $J = \{(i, \varepsilon) \mid i \in I \text{ and } i \notin C\} \cup \{(i, y_i) \mid i \in I \text{ and } i \in C\}$. We only keep in $\mathcal{S}$ the accessible part from the initial state. The transducer $\mathcal{S}$ is sequential. It turns out that the transducer $\mathcal{S}$ has a finite number of states. This will be proved in Lemma 14.

Some definitions are needed to prove the correctness of the construction. We introduce first a distance d on finite words. This distance should not be mixed up with the distance that we have used at the beginning of Section 3 to define the topology on A^ω. For finite words u and v, we denote by d the distance such that

$$d(u,v) = |u| + |v| - 2|u \wedge v|,$$

where $u \wedge v$ is the longest common prefix of u and v (see [4, p. 104]). We extend this distance when v is replaced by an infinite word. Let u be a finite word and let x be an infinite word. We define

$$d(u,x) = |u| - |u \wedge x|,$$

where $u \wedge x$ is the longest common prefix of u and x. In that case, the function d is not a distance but it measures how far u is from being a prefix of x. Note that if u and w are two finite words and if z is a finite or infinite word, the equality $d(wu,wz) = d(u,z)$ holds. The following lemma states some relation between the distance d and the weak twinning property. This is an easy property of combinatorics of words.

Lemma 8 *Let v_1, v_2, v_1' and v_2' be finite words such that $|v_2| = |v_2'|$ and $v_1 v_2^\omega = v_1' {v_2'}^\omega$. For any finite word v_3 and for any finite or infinite word v_3', one has*

$$d(v_1 v_2 v_3, v_1' v_2' v_3') = d(v_1 v_3, v_1' v_3').$$

Proof We first suppose that $|v_1| \leq |v_1'|$. Then there is a finite word w such that $v_1' = v_1 w$ and $w v_2' = v_2 w$. Thus the word $v_1' v_2' v_3'$ is equal to $v_1 v_2 w v_3'$ and it follows that

$$d(v_1 v_2 v_3, v_1' v_2' v_3') = d(v_3, w v_3') = d(v_1 v_3, v_1' v_3').$$

The case where $|v_1| \geq |v_1'|$ can be handled similarly. $\Box$

The transducer $\mathcal{S}$ is sequential but it may not be complete. For a state q and a letter a, there may be no transition going out of q and input labeled by a. For any nonempty finite word u and any states P and P' of $\mathcal{S}$, there is at most one path $P \xrightarrow{u|v} P'$ from P to P'. The following lemma and its corollary state the main property of the transitions of $\mathcal{S}$. This property comes directly from the definition of the transitions of $\mathcal{S}$. No property of $\mathcal{T}$ is assumed.

Lemma 9 *Let u be a nonempty finite word.*

(a) Let $P \xrightarrow{u|v} P'$ be a path from P to P' in $\mathcal{S}$ with input label u. If $(q',z') \in P'$, then there is a pair $(q,z) \in P$ and a path $q \xrightarrow{u|v'} q'$ in $\mathcal{T}$ such that $zv' = vz'$ if $q,q' \notin C$, $zv'y_{q'} = vz'$ if $q \notin C$ and $q' \in C$, and $z = vz'$ if $q,q' \in C$.

(b) *Let P be a state of $\mathcal{S}$. If $(q,z) \in P$ and $q \xrightarrow{u|v'} q'$ is a path in $\mathcal{T}$, then there is a path $P \xrightarrow{u|v} P'$ in $\mathcal{S}$ and a word z' such that $(q',z') \in P'$, $zv' = vz'$ if $q, q' \notin C$, $zv'y_{q'} = vz'$ if $q \notin C$ and $q' \in C$, and $z = vz'$ if $q, q' \in C$.*

Proof We first prove the statement (a). The proof is an easy induction on the length of the word u. If u is a letter, the result follows directly from the definition of the transitions of $\mathcal{S}$. Otherwise, the word u is equal to u_0u_1 where u_0 and u_1 are two nonempty words. The path from P to P' can be factorized

$$P \xrightarrow{u_0|v_0} P'' \xrightarrow{u_1|v_1} P'$$

where $v = v_0v_1$. For each pair (q',z') of P', there are from the induction hypothesis two pairs (q,z) and (q'',z'') in P and P'' and two paths $q \xrightarrow{u_0|v'_0} q''$ and $q'' \xrightarrow{u_1|v'_1} q'$ in $\mathcal{T}$. We discuss on the membership of q, q'' and q' to C.

- If $q \notin C$, $q'' \notin C$ and $q' \notin C$, the induction hypothesis gives $zv'_0 = v_0z''$ and $z''v'_1 = v_1z'$. This implies $zv'_0v'_1 = v_0z''v'_1 = v_0v_1z'$, that is $zv' = vz'$.

- If $q \notin C$, $q'' \notin C$ and $q' \in C$, the induction hypothesis gives $zv'_0 = v_0z''$ and $z''v'_1y_{q'} = v_1z'$. This implies $zv'_0v'_1y_{q'} = v_0z''v'_1y_{q'} = v_0v_1z'$, that is $zv'y_{q'} = vz'$.

- If $q \notin C$, $q'' \in C$ and $q' \in C$, the induction hypothesis gives $zv'_0y_{q''} = v_0z''$ and $z'' = v_1z'$. Since $y_{q''} = v'_1y_{q'}$, this implies that $zv'_0v'_1y_{q'} = zv'_0y_{q''} = v_0z'' = v_0v_1z'$, that is $zv'y_{q'} = vz'$.

- If $q \in C$, $q'' \in C$ and $q' \in C$, the induction hypothesis gives $z = v_0z''$ and $z'' = v_1z'$. This implies $z = v_0v_1z'$, that is $z = vz'$.

The proof of the statement (b) can be handled similarly. □

The following corollary just states the result of the previous lemma when the state P is the initial state J of $\mathcal{S}$.

Corollary 10 *Let u be a nonempty finite word.*

(a) *Let $J \xrightarrow{u|v} P$ be a path from the initial state J to P in $\mathcal{S}$ with input label u. If $(q,z) \in P$, then there is a path $i \xrightarrow{u|v'} q$ in $\mathcal{T}$ such that $v' = vz$ if $q \notin C$, and $v'y_q = vz$ if $q \in C$.*

(b) *If $i \xrightarrow{u|v'} q$ is a path in $\mathcal{T}$, then there is a path $J \xrightarrow{u|v} P$ in $\mathcal{S}$ and a word z such that $(q,z) \in P$, $v' = vz$ if $q \notin C$, and $v'y_q = vz$ if $q \in C$.*

Proof The second component z of a pair (i,z) in J is either the empty word if i is not constant or the word y_i if i is constant. Then the result follows directly from the previous lemma. □

The following four lemmas are devoted to the proof that the transducer $\mathcal{S}$ has finitely many states. It is first proved in the next lemma that in each state P of $\mathcal{S}$ there is at most one occurrence of each state q. Therefore, the number of pairs in each state of $\mathcal{S}$ is bounded by the number of states in $\mathcal{T}$. Then it is proved in the next two lemmas that the lengths of the finite words which appear in the pairs are bounded. It is finally proved in the fourth lemma that the number of infinite words which can appear in the pairs is bounded.

LEMMA 11 *Let $\mathcal{T}$ be a transducer realizing a function f. Let q be a state of $\mathcal{T}$ and let P be a state of $\mathcal{S}$. There is at most one word z such that (q,z) belongs to P.*

Proof Let $J \xrightarrow{u|v} P$ be a path in $\mathcal{S}$ and let (q,z) and (q,z') be two pairs in P.

We first suppose that q is not constant and thus that z and z' are finite. Let $x|y$ and $x'|y'$ be the labels of two final paths starting at q such that $y \neq y'$. By the previous corollary, there are two paths $i \xrightarrow{u|vz} q$ and $i' \xrightarrow{u|vz'} q$ in $\mathcal{T}$. One has $f(ux) = vzy = vz'y$ and $f(ux') = vzy' = vz'y'$. If $z \neq z'$, it may be assumed by symmetry that $|z'| > |z|$ and that $z' = zw$ for some finite word w. This leads to the contradiction $y = y' = w^\omega$.

We now suppose that q is constant and thus that z and z' are infinite. Let $x|y_q$ be the label of a final path starting at q. By the previous corollary, there are two paths $i \xrightarrow{u|w} q$ and $i' \xrightarrow{u|w'} q$ in $\mathcal{T}$ such that $wy_q = vz$ and $w'y_q = vz'$. Furthermore, one has $f(ux) = wy_q = w'y_q$ and thus $z = z'$. $\square$

¿From now on, we always assume that the transducer $\mathcal{T}$ realizes a function f.

LEMMA 12 *Let $\mathcal{T}$ be a transducer which has the weak twinning property. There is a constant K such that for any two paths $i \xrightarrow{u|v} q$ and $i' \xrightarrow{u|v'} q'$ where i and i' are initial states and $q \notin C$, one has*

$$\begin{aligned} d(v,v') &\le K \quad \text{if } q' \notin C \\ d(v,v'y_{q'}) &\le K \quad \text{if } q' \in C \end{aligned}$$

Proof Let K be equal to $2n^2M$ where n is the number of states of the transducer $\mathcal{T}$ and M is the maximal length of the output label of a transition. We prove the inequalities by induction on the length of u. If $|u| \le n^2$, then the result follows easily from $|v|, |v'| \le n^2M$. Otherwise, both paths can be factorized

$$\begin{aligned} &i \xrightarrow{u_1|v_1} p \xrightarrow{u_2|v_2} p \xrightarrow{u_3|v_3} q \\ &i' \xrightarrow{u_1|v_1'} p' \xrightarrow{u_2|v_2'} p' \xrightarrow{u_3|v_3'} q' \end{aligned}$$

where $u_1u_2u_3 = u$, $v_1v_2v_3 = v$, $v_1'v_2'v_3' = v'$, $|u_2| > 0$ and $|u_3| \le n^2$. Since q is not constant, p is also not constant.

We first suppose that p' is not constant. By the weak twinning property and by Lemma 8, one has either $d(v_1v_2v_3, v_1'v_2'v_3') = d(v_1v_3, v_1'v_3')$ if q' is not constant or $d(v_1v_2v_3, v_1'v_2'v_3'y_{q'}) = d(v_1v_3, v_1'v_3'y_{q'})$ otherwise. The result follows from the induction hypothesis.

We now suppose that p' is constant. Therefore, q' is also constant and $y_{p'} = v_3'y_{q'}$ and $y_{p'} = v_2'y_{p'}$. If v_2 is empty, one has $d(v_1v_2v_3, v_1'v_2'v_3'y_{q'}) =$

$d(v_1v_3, v'_1v'_3y_{q'})$ since $v'_2v'_3y_{q'} = v'_3y_{q'}$. The result follows from the induction hypothesis. If v_2 is nonempty, the weak twinning property implies that $v'_1y_{p'} = v_1v_2^\omega$. Therefore, $d(v_1v_2v_3, v'_1v'_2v'_3y_{q'}) \leq |v_3| \leq K$. $\square$

The following lemma states that the lengths of the finite words z of the pairs (q, z) in the states of $\mathcal{S}$ are bounded. It is essentially due to the twinning property of $\mathcal{T}$.

Lemma 13 *Let $\mathcal{T}$ be a transducer which has the weak twinning property. There is a constant K such that for any pair (q, z) in a state P of $\mathcal{S}$, z is infinite if $q \in C$, and $|z| \leq K$ if $q \notin C$.*

Proof Let K be the constant given by the previous lemma. Let (q, z) be a pair in a state P such that the state q of $\mathcal{T}$ is not constant. If (q, z) is the only pair in the state P, the word z must be empty and the result holds. Otherwise, there is another pair (q', z') in P such that z and z' do not have a common prefix. One has $|z| \leq d(z, z') \leq K$. $\square$

It is now possible to prove that the transducer $\mathcal{S}$ has a finite number of states. However, the number of states of $\mathcal{S}$ can be exponential as in the case of finite words.

Lemma 14 *Let $\mathcal{T}$ be a transducer which has the weak twinning property. The number of states of $\mathcal{S}$ is finite.*

Proof We have proved in the preceding lemma that the lengths of the finite words z are bounded. It remains to show that there is a finite number of different infinite words z which can appear in some pair (q, z). By definition of the transitions, any infinite word z of a pair is the suffix of $z'wy_p$ where (p', z') is a pair such that $p' \notin C$ and z' is finite and where $p \in C$ and $p' \xrightarrow{a|w} p$ is a transition of $\mathcal{T}$. Since the length of z' is bounded, the number of such words $z'wy_p$ is finite and they are ultimately periodic. Then there are a finite number of suffixes of such words. $\square$

The following lemma states the key property of $\mathcal{S}$. Its purpose is to guarantee that the transducer $\mathcal{S}$ has the same output as $\mathcal{T}$ up to a bounded suffix.

Lemma 15 *Let $\mathcal{T}$ be a transducer satisfying the conditions of Theorem 3 and let $\mathcal{S}$ be the corresponding sequential transducer. Let $q \xrightarrow{u|v} q$ and $P \xrightarrow{u|v'} P$ be cycling paths in $\mathcal{T}$ and $\mathcal{S}$ where the state P contains a pair (q, z). If the path $q \xrightarrow{u|v} q$ contains a final state and if v is nonempty, then v' is also nonempty.*

Proof By Lemma 11, there is only one word z such that (q, z) belongs to P. Since the state P is accessible, there is a path $J \xrightarrow{t|w'} P$ in $\mathcal{S}$. By Corollary 10, there is a path $i \xrightarrow{t|w} q$ in $\mathcal{T}$ for some finite word w. The paths are summarized by the following diagram.

$$\begin{array}{ccccc} i & \xrightarrow{t|w} & q & \xrightarrow{u|v} & q \\ J & \xrightarrow{t|w'} & P & \xrightarrow{u|v'} & P \end{array}$$

We assume that the loop $q \xrightarrow{u|v} q$ around q contains a final state. Since $v \neq \varepsilon$, the word tu^ω belongs to the domain of the function and $f(tu^\omega) = wv^\omega$.

We distinguish two main cases depending on whether q is a constant state or not. In the case that q is not constant, the hypothesis that the path $q \xrightarrow{u|v} q$ goes through a final state is not needed. This fact is used in the proof of the other case.

We first suppose that q is not constant. The word z is thus finite. By Corollary 10 applied to the paths $J \xrightarrow{t|w'} P$ and $J \xrightarrow{tu|w'v'} P$, both equalities $w = w'z$ and $wv = w'v'z$ hold. This implies that $|v'| = |v|$ and the word v' is nonempty.

We now suppose that q is constant. The word z is thus infinite. We distinguish again two cases depending on whether the state P contains at least a nonconstant state or not. In both cases, we use the following claim. For any pair (q', z') in P, there is a pair (q'', z'') in P such that there are paths in $\mathcal{T}$ and $\mathcal{S}$ as shown in the following diagram

$$\begin{array}{ccccccc} i' & \xrightarrow{t|w''} & q'' & \xrightarrow{u^k|v''} & q'' & \xrightarrow{u^l|v'''} & q' \\ J & \xrightarrow{t|w'} & P & \xrightarrow{u^k|v'^k} & P & \xrightarrow{u^k|v'^l} & P \end{array}$$

where k is a positive integer and l is a nonnegative integer. Let (q', z') be any pair in P. Define by induction the sequence $(q_n, z_n)_{n\geq 0}$ of pairs in P as follows. Let (q_0, z_0) be the pair (q', z'). Suppose that the pair (q_n, z_n) is already defined. By Lemma 9, there is a pair (q_{n+1}, z_{n+1}) in P such that there is a path $q_{n+1} \xrightarrow{u|w_n} q_n$ in $\mathcal{T}$. Since the set P is finite, there are two integers $k \geq 1$ and $l \geq 0$ such that $q_{k+l} = q_l$ and thus $z_{k+l} = z_l$ by Lemma 11. Let (q'', z'') denote the pair (q_l, z_l). By construction of q'', there is in $\mathcal{T}$ a cycling path $q'' \xrightarrow{u^k|v''} q''$ and there is also a path $q'' \xrightarrow{u^l|v'''} q'$. Since the pair (q'', z'') belongs to P, there is, by Corollary 10, a path $i' \xrightarrow{t|w''} q''$ in $\mathcal{T}$, with $i' \in I$. This proves the claim.

We first suppose that P contains a pair (q', z') such that q' is not constant. Let (q'', z'') be the pair given by the previous claim. Since there is a path from q'' to q', the state q'' is also not constant. We prove by contradiction that v'' is nonempty. Let us assume that $v'' = \varepsilon$. Since q'' is not constant, there are two final paths starting at q'' with different output labels. Let $x|y$ and $x'|y'$ be the labels of these two final paths with $y \neq y'$. The images $f(tu^{kn}x)$ and $f(tu^{kn}x')$ are equal to $w''y$ and $w''y'$ for any integer n. Both sequences $(tu^{kn}x)_{n\geq 0}$ and $(tu^{kn}x')_{n\geq 0}$ converge to tu^ω. Since the function f is continuous, both words $w''y$ and $w''y'$ are equal to $f(tu^\omega) = wv^\omega$. This is a contradiction since $y \neq y'$. This proves that $v'' \neq \varepsilon$. Since q'' is not constant, the proof of the first case can be applied to the paths $q'' \xrightarrow{u^k|v''} q''$ and $P \xrightarrow{u^k|v'^k} P$. This proves that v'^k and thus v' is nonempty.

We finally suppose that for every pair (q', z') in P, the state q' is constant. Let (q', z') be any pair in P and let (q'', z'') be the pair given by the claim above. We prove that $z = z''$. By hypothesis, the state q'' is constant. Let $x|y_{q''}$ be the label of a final path starting at q''. Since q'' is constant, the equality $v''y_{q''} = y_{q''}$ holds. The image $f(tu^{kn}x)$ is equal to $w''v''^n y_{q''} = w''y_{q''}$ for any integer n. The sequence $(tu^{kn}x)_{n\geq 0}$ converges to tu^ω. Since the function f is continuous, the word $w''y_{q''}$ is equal to $f(tu^\omega) = wv^\omega = wy_q$. By Corollary 10 applied to the path $J \xrightarrow{t|w'} P$, both equalities $wy_q = w'z$ and $w''y_{q''} = w'z''$ hold. Combined with the equality $w''y_{q''} = wy_q$, one gets $z = z''$.

By Lemma 9 applied to the path $P \xrightarrow{u^l|v'^l} P$, the equality $z'' = v'^l z'$ holds. If $v' = \varepsilon$, then $z'' = z = z'$. Since this equality holds for any pair (q', z') of P, all words z' of the pairs (q', z') in P are equal. This contradicts the definition of the transitions of $\mathcal{S}$ since the output v' along the path $P \xrightarrow{u|v'} P$ is nonempty in this case. This implies that $v' \neq \varepsilon$. □

The following proposition states that the function realized by the sequential transducer $\mathcal{S}$ is an extension of the function realized by the transducer $\mathcal{T}$.

Proposition 16 *Let $\mathcal{T}$ be a transducer satisfying the conditions of Theorem 3 and let $\mathcal{S}$ be the corresponding sequential transducer. Let f and f' be the functions realized by the transducers $\mathcal{T}$ and $\mathcal{S}$. Then the inclusion* $\mathrm{dom}(f) \subseteq \mathrm{dom}(f')$ *holds and for any x in* $\mathrm{dom}(f)$, *the equality $f(x) = f'(x)$ holds.*

Proof We prove that if the infinite word x belongs to the domain of f, it also belongs to the domain of f' and its images by f and f' are equal.

Let x be an infinite word which belongs to the domain of f and let γ be a successful path in $\mathcal{T}$ with input label x. Therefore, this path goes infinitely often through a final state and its output label is an infinite word. Consider the unique path Γ in $\mathcal{S}$ with input label x.

We claim that the output label along Γ is nonempty and that it is equal to the output label along γ. Since both transducers $\mathcal{T}$ and $\mathcal{S}$ (by Lemma 14) have a finite number of states, both paths γ and Γ can be factorized

$$\begin{aligned} \gamma &= i \xrightarrow{u_0|v_0} q \xrightarrow{u_1|v_1} q \xrightarrow{u_2|v_2} q \cdots \\ \Gamma &= J \xrightarrow{u_0|v'_0} P \xrightarrow{u_1|v'_1} P \xrightarrow{u_2|v'_2} P \cdots \end{aligned}$$

Since the output along the path γ is infinite, it can be assumed that each word v_n is nonempty and since the path γ goes infinitely often through a final state, it can be also assumed that each path $q \xrightarrow{u_n|v_n} q$ contains a final state. By Corollary 10, the state P of $\mathcal{S}$ contains a pair (q, z) for some finite or infinite word z. By Lemma 15, each word v'_n is nonempty.

By Corollary 10, one has for each n, $v_0 \ldots v_n = v'_0 \ldots v'_n z$ if q is not constant and one has $v_0 \ldots v_n y_q = v'_0 \ldots v'_n z$ otherwise. This implies the equality $v_0 v_1 v_2 \ldots = v'_0 v'_1 v'_2 \ldots$ of the two outputs. □

By the last proposition, the function realized by the sequential transducer $\mathcal{S}$ extends the function realized by the given transducer $\mathcal{T}$. To obtain a sequential transducer equivalent to $\mathcal{T}$, one must restrict the domain of the transducer $\mathcal{S}$. This is achieved by constructing a new sequential transducer $\mathcal{S}'$ which is the synchronized product of $\mathcal{S}$ and of an automaton for the domain of $\mathcal{T}$.

Recall that the transducer $\mathcal{S}$ has no acceptance condition. This means that an infinite path is final iff both its input and output labels are infinite words.

Let X be the domain of the function realized by $\mathcal{T}$. Let $\mathcal{A}$ be a deterministic Büchi automaton recognizing X if X is deterministic or let $\mathcal{A}$ be a deterministic Muller automaton recognizing X otherwise. In the former case, its acceptance condition Φ is a set F of final states and, in the latter case, its acceptance condition Φ is a family $\mathcal{F}$ of sets of states. As explained at the end of Section 2, the automaton $\mathcal{A}$ can be computed from the transducer $\mathcal{T}$.

We now describe the transducer $\mathcal{S}'$. The state set of $\mathcal{S}'$ is $Q \times Q'$ where Q and Q' are the state sets of $\mathcal{S}$ and $\mathcal{A}$. The initial state is (i, i') where i and i' are the initial states of $\mathcal{S}$ and $\mathcal{A}$. There is a transition $(p, p') \xrightarrow{a|u} (q, q')$ iff $p \xrightarrow{a|u} q$ and $p' \xrightarrow{a} q'$ are transitions of $\mathcal{S}$ and $\mathcal{A}$. The acceptance condition Φ' of $\mathcal{S}'$ mimics that of $\mathcal{A}$. More formally, if $\mathcal{A}$ is a Büchi automaton, then $\mathcal{S}'$ is a Büchi transducer and its set of final states is $F' = \{(q, q') \mid q' \in F\}$. If $\mathcal{A}$ is a Muller automaton, then $\mathcal{S}'$ is a Muller transducer and its family $\mathcal{F}'$ of sets of states is defined as follows.

$$\mathcal{F}' = \{\{(q_1, q_1'), \ldots, (q_k, q_k')\} \mid \{q_1', \ldots, q_k'\} \in \mathcal{F}\}.$$

It is pure routine to check that $\mathcal{S}'$ is equivalent to $\mathcal{T}$.

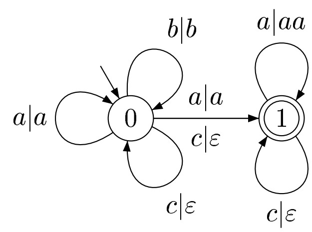


Figure 5: Transducer $\mathcal{T}$ of Example 17

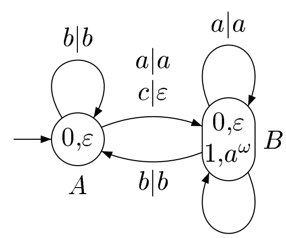


Figure 6: Transducer $\mathcal{S}$ of Example 17

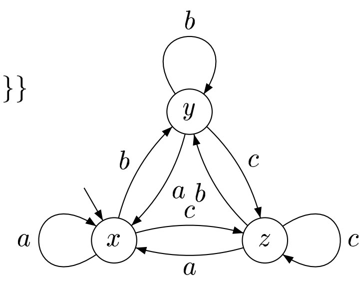


Figure 7: A Muller automaton for the domain of $\mathcal{T}$

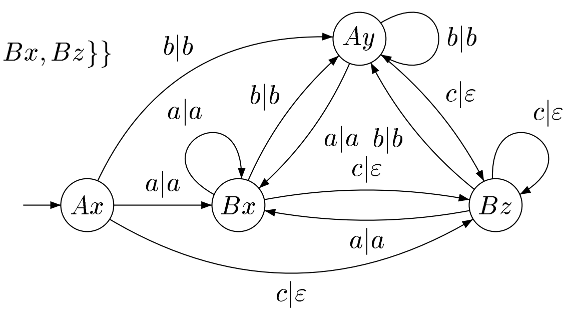


Figure 8: Transducer $\mathcal{S}'$ of Example 17

We illustrate the construction of $\mathcal{S}$ and $\mathcal{S}'$ by the following example.

Example 17 Let A be the alphabet $\{a, b, c\}$ and consider the transducer $\mathcal{T}$ pictured in Figure 5. Note that the state 1 is constant whereas the state 0 is not. Applying the construction described above, one gets the transducer $\mathcal{S}$ pictured in Figure 6. The domain of $\mathcal{T}$ is $A^*(c^*a)^\omega$ but the domain of $\mathcal{S}$ is $A^*(c^*(a+b))^\omega$. The Muller automaton $\mathcal{A}$ for the domain of $\mathcal{T}$ is pictured in Figure 7. The transducer $\mathcal{S}'$ obtained by combining $\mathcal{S}$ and $\mathcal{A}$ is pictured in Figure 8.

5 Conclusion

In this paper, we have provided characterizations of sequential functions of infinite words realized by Muller and Büchi transducers. When a transducer realizes a sequential function, we have given an algorithm to compute an equivalent sequential transducer. Since this determinization includes the determinization of an automaton for the domain of the function, the complexity is at least exponential.

In the case of finite words, the determinization is also exponential but it can be checked in polynomial time whether a function given by a transducer is sequential. The continuity can be checked in polynomial time [15]. The decidability of the weak twinning property that we introduce is not discussed in the paper. We do not know whether this can be checked in polynomial time. However, since this notion is close to the twinning property of Choffrut [6, 7], we think that the methods used in [22] or [3] can be used to obtain a polynomial time algorithm to check this property.

Acknowledgments

The authors would like to thank Joost Engelfriet and the anonymous referees for their highly relevant comments and very helpful suggestions about the preliminary version of this paper.

References

[1] Béal, M.-P., and Carton, O. Computing the prefix of an automaton. *RAIRO Theoret. Inform. Appl. 34*, 6 (2000), 503–514.

[2] Béal, M.-P., and Carton, O. Determinization of transducers over finite and infinite words. *Theoret. Comput. Sci. 289*, 1 (2002), 225–251.

[3] Béal, M.-P., Carton, O., Prieur, C., and Sakarovitch, J. Squaring transducers: An efficient procedure for deciding functionality and sequentiality. *Theoret. Comput. Sci. 292*, 1 (2003), 45–63.

[4] Berstel, J. *Transductions and Context-Free Languages.* B.G. Teubner, 1979.

[5] Breslauer, D. The suffix tree of a tree and minimizing sequential transducers. *Theoret. Comput. Sci.*, 191 (1998), 131–144.

[6] Choffrut, C. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoret. Comput. Sci. 5* (1977), 325–338.

[7] Choffrut, C. *Contribution à l'étude de quelques familles remarquables de fonctions rationnelles.* Thèse d'État, Université Paris VII, 1978.

[8] Cohen, A., and Collard, J.-F. Instance-wise reaching definition analysis for recursive programs using context-free transductions. In *Parallel Architectures and Compilation Techniques* (1998), IEEE Computer Society Press, pp. 332–340.

[9] Gire, F. Two decidability problems for infinite words. *Inform. Proc. Letters 22* (1986), 135–140.

[10] Lind, D., and Marcus, B. *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, 1995.

[11] Lothaire, M. *Algebraic Combinatorics on Words.* Cambridge University Press, 2002, ch. 7, pp. 230–268.

[12] McNaughton, R. Testing and generating infinite sequences by a finite automaton. *Inform. Control 9* (1966), 521–530.

[13] Mohri, M. On some applications of finite-state automata theory to natural languages processing. *Journal of Natural Language Engineering 2* (1996), 1–20.

[14] Mohri, M. Minimization algorithms for sequential transducers. *Theoret. Comput. Sci.*, 234 (2000), 177–201.

[15] Prieur, C. How to decide continuity of rational functions on infinite words. *Theoret. Comput. Sci. 250* (2001), 71–82.

[16] Roche, E., and Schabes, Y. *Finite-State Language Processing.* MIT Press, Cambridge, 1997, ch. 7.

[17] Safra, S. On the complexity of ω-automata. In *29th Annual Symposium on Foundations of Computer Science* (1988), pp. 319–327.

[18] Sakarovitch, J. *Éléments de Théorie des Automates.* Vuibert, 2003. To appear.

[19] Schützenberger, M.-P. Sur les relations rationnelles. In *Automata Theory and Formal Languages, 2nd GI Conference* (1975), H. Brakhage, Ed., vol. 33 of *Lect. Notes in Comput. Sci.*, Springer, pp. 209–213.

[20] Thomas, W. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, J. van Leeuwen, Ed., vol. B. Elsevier, 1990, ch. 4, pp. 133–191.

[21] Thomas, W. Languages, automata and logic. In *Handbook of Formal Languages*, G. Rozenberg and A. Salomaa, Eds., vol. 3. Springer-Verlag, 1997, pp. 389–449.

[22] Weber, A., and Klemm, R. Economy of description for single-valued transducers. *Inform. Comput. 118* (1995), 327–340.