Determinization of transducers over infinite words

Marie-Pierre Béal[1] and Olivier Carton[2]

[1] Institut Gaspard Monge,
Université de Marne-la-Vallée
Marie-Pierre.Beal@univ-mlv.fr
http://www-igm.univ-mlv.fr/~beal/
[2] Institut Gaspard Monge and CNRS
Université de Marne-la-Vallée
Olivier.Carton@univ-mlv.fr
http://www-igm.univ-mlv.fr/~carton/

Abstract. We study the determinization of transducers over infinite words. We consider transducers with all their states final. We give an effective characterization of sequential functions over infinite words. We also describe an algorithm to determinize transducers over infinite words.

1 Introduction

The aim of this paper is the study of determinization of transducers over infinite words, that is of machines realizing rational transductions. A transducer is a finite state automaton (or a finite state machine) whose edges are labeled by pairs of words taken in finite alphabets. The first component of each pair is called the input label. The second one the output label. The rational relation defined by a transducer is the set of pairs of words which are labels of an accepting path in the transducer. We assume that the relations defined by our transducers are functions which each string of the domain to a string. This is a decidable property [8].

The study of transducers has many applications. Transducers are used to model coding schemes (compression schemes, convolutional coding schemes, coding schemes for constrained channels, for instance). They are widely used in computer arithmetic [7] and in natural language processing [13]. Transducers are also used in programs analysis [6]. The determinization of a transducer is the construction of another transducer which defines the same function and has a deterministic (or right resolving) input automaton. Such transducers allow a sequential encoding and thus are called sequential transducers.

The characterization of sequential functions on finite words was obtained by Choffrut [4, 5]. His proof contains implicitly an algorithm for determinization of a transducer. This algorithm has also been described by Mohri [11] and Roche and Shabes [13, p. 223–233]. In this paper, we address the same problem for infinite words. We consider transducers and functions over infinite words and our transducers have all their states final. The reason why we assume that all states are final is that the case of transducers with final states seems to be much more complex. Indeed, the determinization of automata over infinite words is already very difficult [14]. In particular, it is not true that any rational set of infinite words is recognized by a deterministic automaton with final states and Büchi acceptance condition. Other accepting conditions, as the Muller condition for instance, must be used.

We first give an effective characterization of sequential functions over infinite words. This characterization extends to infinite words the twinning property introduced by Choffrut [4]. We prove that a function is sequential if it is a continuous map whose domain can be recognized by a deterministic Büchi automaton, and such that the transducer obtained after removing some special states has the twinning property. These conditions can be simplified in the case where the transducer has no cycling path with an empty output label. We use this characterization to describe an algorithm checking whether a function realized by a transducer is sequential. This algorithm becomes polynomial when the transducer has no cycling path with an empty output label. Finally, we give an algorithm to determinize a transducer. The algorithm is much more complex than in the case of finite words.

The paper is organized as follows. Section 2 is devoted to basic notions of transducers and rational functions. We give in Sect. 3 a characterization of sequential functions while the algorithm for determinization of transducers is described in Sect. 4.

2 Transducers

In the sequel, A and B denote finite alphabets. The set of finite and right-infinite words over A are respectively denoted by A^* and A^ω. The empty word is denoted by ε. The set A^ω is endowed with the usual topology induced by the following metric: the distance $d(x, y)$ is equal 2^{-n} where n is the minimum $\min\{i \mid x_i \neq y_i\}$. In this paper, a function from A^ω to B^ω is said to be continuous iff it is continuous with respect to this topology.

A *transducer* over $A \times B$ is composed of a finite set Q of *states*, a set $E \subseteq Q \times A^* \times B^* \times Q$ of *edges* and a set $I \subseteq Q$ of *initial* states. 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*. Thus, a transducer is the same object as an automaton, except that the labels of the edges are pairs of words instead of letters. In the literature, transducers also have a set of final states. In this paper, we only consider transducers all of which states are final and with Büchi acceptance condition. Any infinite path which starts at an initial state is then successful. We omit the set of final states in the notation.

An infinite *path* in the transducer $\mathcal{A}$ 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 $x = u_0u_1u_2\ldots$ whereas its output label is the word $y = v_0v_1v_2\ldots$. The path is said to *start* at q_0.

An infinite path is then *successful* if it starts at an initial state. A pair (x, y) of infinite words is *recognized* by the transducer if it labels a successful path. A transducer defines then a relation $R \subseteq A^\omega \times B^\omega$. The transducer computes a function if for any word $x \in A^\omega$, there exists at most one word $y \in B^\omega$ such that $(x, y) \in R$. We call it the function *realized* by the transducer. Thus a transducer can be seen as a machine computing nondeterministically output words from input words. We denote by $\mathrm{dom}(f)$ the domain of the function f. A transducer that realizes a function can be transformed in an effective way in a transducer labelled in $A \times B^*$ that realizes the same function. These transducers are sometimes called *real time transducers.*

Let $\mathcal{T}$ be a transducer. The *underlying input automaton* of $\mathcal{T}$ is obtained by omitting the output label of each edge. A transducer $\mathcal{T}$ is said to be *sequential* if it is labeled in $A \times B^*$ and if the following conditions are satisfied.

- it has a unique initial state,
- the underlying input automaton is deterministic.

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 computed by $\mathcal{T}$ is a partial function from A^ω into B^ω. A function is *sequential* if it can be realized by a sequential transducer.

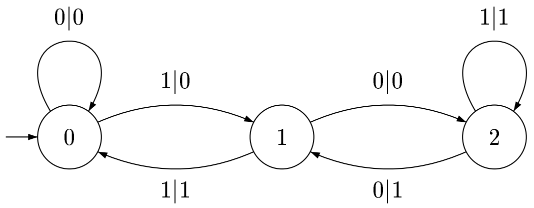

Fig. 1. Transducer of Example 1

Example 1. Let $A = \{0, 1\}$ be the binary alphabet. Consider the sequential transducer $\mathcal{T}$ pictured in Fig. 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 $\alpha/3$. The transducer $\mathcal{T}$ realizes the division by 3 on binary expansions. The transducer obtained by exchanging the input and output labels of each edge realizes of course the multiplication by 3. However, this new transducer is not sequential.

3 Characterization of Sequential Functions

In this section, we characterize functions realized by transducers with all states final that can be realized by sequential transducers. This characterization uses topological properties of the function and some twinning property of the transducer. It extends the result of Choffrut [4, 5] to infinite words.

The characterization of the sequentiality is essentially based on the following notion introduced by Choffrut [5, p. 133] (see also [3, p. 128]). Two states q and q' of a transducer are said to be *twinned* if and only if for any pair of paths

$$i \xrightarrow{u|u'} q \xrightarrow{v|v'} q$$
$$i' \xrightarrow{u|u''} q' \xrightarrow{v|v''} q',$$

where i and i' are two initial states, the output labels satisfy the following property. Either $v' = v'' = \varepsilon$ or there exists a finite word w such that either $u'' = u'w$ and $wv'' = v'w$, or $u' = u''w$ and $wv' = v''w$. The latter case is equivalent to the following two conditions:

(i) $|v'| = |v''|$,
(ii) $u'v'^{\omega} = u''v''^{\omega}$

A transducer has the *twinning property* if any two states are twinned.

Before stating the main result, we define a subset of states which play a particular role in the sequel. We say that a state q of a transducer is *constant* if all infinite paths starting at this state have the same output label. This unique output is an ultimately periodic word. It should be noticed that any state accessible from a constant state is also constant. We now state the characterization of sequential functions.

Proposition 1. *Let f be a function realized by a transducer $\mathcal{T}$. Let $\mathcal{T}'$ be the transducer obtained by removing from $\mathcal{T}$ all states which are constant. Then the function f is sequential if and only if the following three properties hold:*

- *the domain of f can be recognized by a deterministic Büchi automaton,*
- *the function f is continuous,*
- *the transducer $\mathcal{T}'$ has the twinning property.*

Since the function f is realized by a transducer, the domain of f is rational. However, it is not true that any rational set of infinite words is recognized by a deterministic Büchi automaton. Landweber's theorem states that a set of infinite words is recognized by a deterministic Büchi automaton if and only if it is rational and G_δ [16]. Recall that a set is said to be G_δ is it is equal to a countable union of open sets for the usual topology of A^ω.

It is worth pointing out that the domain of a function realized by a transducer may be any rational set although it is supposed that all states of the transducer are final. The final states of a Büchi automaton can be encoded in the outputs of a transducer in the following way. Let $\mathcal{A} = (Q, E, I, F)$ be a Büchi automaton.

We construct a transducer $\mathcal{T}$ by adding an output to any transition of $\mathcal{A}$. A transition $p \xrightarrow{a} q$ of $\mathcal{A}$ becomes $p \xrightarrow{a|v} q$ in $\mathcal{T}$ where v is empty if p is not final and is equal to a fixed letter b if p is final. It is clear that the output of a path is infinite if and only if the path goes infinitely often through a final state. Thus the domain of the transducer $\mathcal{T}$ is the set recognized by $\mathcal{A}$. For instance, the domain of a transducer may be not recognizable by a deterministic Büchi automaton as in the following example. It is however true that the domain is closed if the transducer has no cycling path with an empty output.

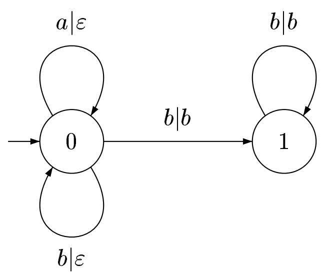

Fig. 2. Transducer of Example 2

Example 2. The domain of the function f realized by the transducer of Fig. 2 is the set $(a+b)^*b^\omega$ of words having a finite number of a. The function f cannot be realized by a sequential transducer since its domain is not a G_δ set.

It must be also pointed out that a function realized by a transducer may be not continuous although it is supposed that all states of the transducer are final as it is shown in the following example.

Example 3. The image of an infinite word x by the function f realized by the transducer of Fig. 3 is $f(x) = a^\omega$ if x has an infinite number of a and $f(x) = a^n b^\omega$ if the number of a in x is n. The function f is not continuous. For instance, the sequence $x_n = b^n a b^\omega$ converges to b^ω while $f(x_n) = ab^\omega$ does not converge to $f(b^\omega) = b^\omega$.

Before describing the algorithm for determinization, we first study a particular case. It turns out that the first two conditions of the proposition are due to the fact that the transducer $\mathcal{T}$ may have cycling paths with an empty output. If the transducer $\mathcal{T}$ has no cycling path with an empty output, the previous proposition can be stated in the following way.

Proposition 2. *Let f be a function realized by a transducer $\mathcal{T}$ which has no cycling path with an empty output. Let $\mathcal{T}'$ be the transducer obtained by removing*

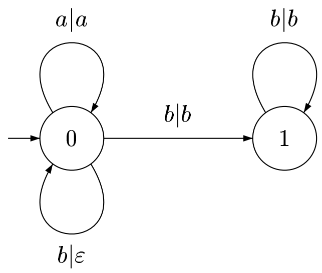

Fig. 3. Transducer of Example 3

from $\mathcal{T}$ all states which are constant. Then the function f is sequential if and only if the transducer $\mathcal{T}'$ has the twinning property.

The previous proposition can be directly deduced from Proposition 1 as follows. If the transducer $\mathcal{T}$ has no cycling path with an empty output, any infinite path has an infinite output. Thus, an infinite word x belongs to the domain of f if and only if it is the input label of an infinite path in $\mathcal{T}$. The domain of f is then a closed set. It is then recognized by a deterministic Büchi automaton whose all states are final. This automaton can be obtained by the usual subset construction on the input automaton of $\mathcal{T}$. Furthermore, if the transducer $\mathcal{T}$ has no cycling path with an empty output, the function f is necessarily continuous.

We now study the decidability of the conditions of Propositions 1 and 2. We have the following results.

Proposition 3. *It is decidable if a function f given by a transducer with all states final is sequential. Furthermore, if the transducer has no cycling path with an empty output, this can be decided in polynomial time.*

A Büchi automaton recognizing the domain of the function can be easily deduced from the transducer. It is then decidable if this set can be recognized by a deterministic Büchi automaton [16, thm 5.3]. However, this decision problem is NP-complete.

It is decidable in polynomial time whether a function given by a transducer with final states is continuous [12]. The twinning property of a transducer is decidable in polynomial time [2].

4 Determinization of Transducers

In this section, we describe an algorithm to determinize a transducer which satisfies the properties of Proposition 1. This algorithm proves that the conditions of the proposition are sufficient. The algorithm is exponential in the number of states of the transducer.

Let $\mathcal{T} = (Q, E, I)$ be a transducer labelled in $A \times B^*$ that realizes a function f. Let $\mathcal{T}'$ be the transducer obtained by removing from $\mathcal{T}$ all states which are constant. We assume that $\mathcal{T}'$ has the twinning property. We denote by C the set of states which are constant. For a state q of C, we denote by y_q, the unique output of q which is an ultimately periodic word. We suppose that the domain of f is recognized by the deterministic Büchi automaton $\mathcal{A}$. This automaton is used in the constructed transducer to ensure that the output is infinite only when the input belongs to the domain of the function.

We describe the deterministic transducer $\mathcal{D}$ realizing the function f. Roughly speaking, this transducer is the synchronized product of the automaton $\mathcal{A}$ of the domain and of an automaton obtained by a variant of the subset construction applied on the transducer. In the usual subset contruction, a state of the deterministic automaton is a subset of states which memorizes all accessible states. In our variant of the subset construction, a state is a subset of pairs formed of a state and a word which is either finite of infinite.

A state of $\mathcal{D}$ is a pair (p, P) where p is a state of $\mathcal{A}$ and P is a set containing two kinds of pairs. The first kind are pairs (q, z) where q belong 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 $\mathcal{D}$. Let (p, P) be a state of $\mathcal{D}$ and let a be a letter. Let R be equal to the set defined as follows

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

We now define the transition from the state (p, P) input labeled by a. If R is empty, there is no transition from (p, P) input labeled by a. Otherwise, the output of this transition is the word v defined as follows. Let $p \xrightarrow{a} p'$ be the transition in $\mathcal{A}$ from p labeled by a. If p' is not a final state of $\mathcal{A}$, we define v as the empty word. If p' is a final state, we define v as the first letter of the words z if R only contains pairs (q', z) with $q' \in C$ and if 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 then defined as follows

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

There is then a transition $(p, P) \xrightarrow{a|v} (p', P')$ in $\mathcal{D}$. The initial state of $\mathcal{D}$ is the pair $(i_{\mathcal{A}}, J)$ where $i_{\mathcal{A}}$ is the initial state of $\mathcal{A}$ and where $J = \{(i, \varepsilon) \mid i \in I \text{ and } i \notin C\} \cup \{(i, y_i) \mid i \in I \text{ and } i \in C\}$. If the state p' is not final in $\mathcal{A}$, the output of the transition from (p, P) to (p', P') is empty and the words z of the pairs (q, z) in P, may have a nonempty common prefix. We only keep in $\mathcal{D}$ the accessible part from the initial state. The transducer $\mathcal{D}$ has a deterministic input automaton. It turns out that the transducer $\mathcal{D}$ has a finite number of states.

The following proposition finally states that the sequential transducer $\mathcal{D}$ is finite and that it is equivalent to the transducer $\mathcal{T}$. Both transducers realize the same function over infinite words.

Proposition 4. *The sequential transducer $\mathcal{D}$ has a finite number of states and it realizes the same function f as the transducer $\mathcal{T}$.*

It is not straightforward that the transducer $\mathcal{D}$ has actually a finite number of states. It must be proved that the finite words which occur as second component of the pairs in the states are bounded. It follows then that the infinite words occuring as second component of the pairs are suffixes of a finite number of ultimately periodic words. Therefore, there are finitely many such words.

It must also be proved that the transducer $\mathcal{D}$ realizes the same function as $\mathcal{T}$. This follows mainly from the following lemma which states the key property of the edges in $\mathcal{D}$.

Lemma 1. *Let u be a finite word. Let $(i_\mathcal{A}, J) \xrightarrow{u|v} (p, P)$ be the unique path in $\mathcal{D}$ with input label u from the initial state. Then, the state p is the unique state of $\mathcal{A}$ such that $i_\mathcal{A} \xrightarrow{u} p$ is a path in $\mathcal{A}$ and the set P is equal to*

$$\begin{aligned} P = \{(q,z) \mid \exists\, i \xrightarrow{u|v'} q \text{ in } \mathcal{T} \text{ such that } v' &= vz \text{ if } q \notin C \\ v'y_q &= vz \text{ if } q \in C\} \end{aligned}$$

This construction is illustrated by the following example.

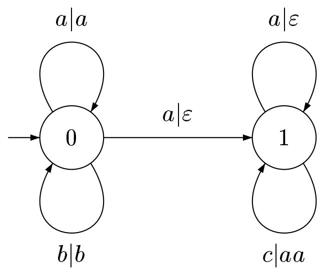

Fig. 4. Transducer of Example 4

Example 4. Consider the transducer pictured in Fig. 4. A deterministic Büchi automaton recognizing the domain is pictured in Fig. 5. If the algorithm for determinization is applied to this transducer, one gets the transducer pictured in Fig. 6.

These determinizations do not preserve the dynamic properties of the transducers as the locality of its output automaton. Recall that a finite automaton is *local* if any two biinfinite paths with the same label are equal. We mention that in [9], an algorithm is given to determinize transducers over bi-infinite words

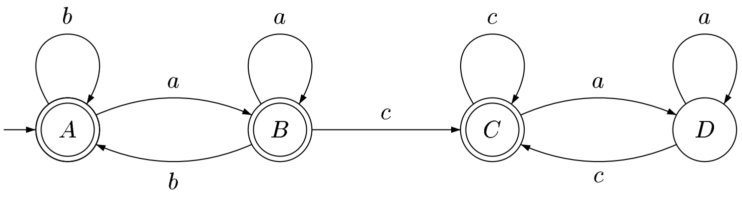

Fig. 5. A deterministic Büchi automaton for the domain

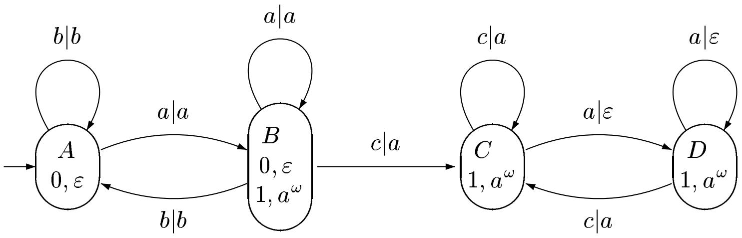

Fig. 6. Determinization of the transducer of Fig. 4

that have a right closing input (or that are n-deterministic or deterministic with a finite delay in the input) and a local output (see also [10, p. 143] and [1, p. 110–115]). This algorithm preserves the locality of the output. These features are important for coding applications.

Acknowledgments

The authors would like to thank Jean Berstel for very helpful suggestions and Christian Choffrut and Isabelle Fagnot for their relevant comments on a preliminary version of this paper.

References

1. Marie-Pierre Béal. *Codage Symbolique.* Masson, 1993.
2. Marie-Pierre Béal, Olivier Carton, Chrhistophe Prieur, and Jacques Sakarovitch. Squaring transducers. In *LATIN'2000.*
3. Jean Berstel. *Transductions and Context-Free Languages.* B.G. Teubner, 1979.
4. Christian Choffrut. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoret. Comput. Sci.*, 5:325–338, 1977.
5. Christian Choffrut. *Contribution à l'étude de quelques familles remarquables de fonctions rationnelles.* Thèse d'État, Université Paris VII, 1978.
6. A. Cohen and J.-F. Collard. Instance-wise reaching definition analysis for recursive programs using context-free transductions. In *PACT'98*, 1998.
7. Christiane Frougny. Numeration systems. In M. Lothaire, editor, *Algebraic Combinatorics on Words.* Cambridge, 1999. to appear.

8. F. Gire. Two decidability problems for infinite words. *Inform. Proc. Letters*, 22:135–140, 1986.
9. Bruce Kitchens. *Continuity properties of factor maps in ergodic theory*. Ph.D. thesis, University of North Carolina, Chapel Hill, 1981.
10. Doug Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.
11. Mehryar Mohri. On some applications of finite-state automata theory to natural languages processing. *Journal of Natural Language Engineering*, 2:1–20, 1996.
12. Christophe Prieur. How to decide continuity of rational functions on infinite words. *Theoret. Comput. Sci.*, 1999.
13. Emmanuel Roche and Yves Schabes. *Finite-State Language Processing*, chapter 7. MIT Press, Cambridge, 1997.
14. Shmuel Safra. On the complexity of ω-automata. In *29th Annual Symposium on Foundations of Computer Sciences*, pages 24–29, 1988.
15. Ludwig Staiger. Sequential mappings of ω-languages. *RAIRO-Infor. Théor. et Appl.*, 21(2):147–173, 1987.
16. Wolfgang Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 4, pages 133–191. Elsevier, 1990.