



**HAL**  
open science

## Lazard's Elimination (in traces) is finite-state recognizable

G rard Henry Edmond Duchamp, Jean-Gabriel Luque

► **To cite this version:**

G rard Henry Edmond Duchamp, Jean-Gabriel Luque. Lazard's Elimination (in traces) is finite-state recognizable. 2006. hal-00084721v2

**HAL Id: hal-00084721**

**<https://hal.science/hal-00084721v2>**

Preprint submitted on 12 Jul 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destin e au d p t et   la diffusion de documents scientifiques de niveau recherche, publi s ou non,  manant des  tablissements d'enseignement et de recherche fran ais ou  trangers, des laboratoires publics ou priv s.

# Lazard's elimination (in traces) is finite-state recognizable

G erard DUCHAMP<sup>†</sup> and Jean-Gabriel LUQUE<sup>‡</sup>

<sup>‡</sup>Laboratoire d'Informatique de Paris Nord Universit  Paris 13  
(LIPN UMR 7030 du CNRS),  
Avenue J.-B. Cl ment 93430 Villetaneuse, France  
Gerard.Duchamp@lipn.univ-paris13.fr

<sup>†</sup> Laboratoire d'informatique de l'Institut Gaspard Monge  
(UMR 8049 IGM-LabInfo ),  
Universit  de Marne-la-Vall e,  
5, Boulevard Descartes,  
775454 Champs-sur-Marne, France  
Jean-Gabriel.Luque@univ-mlv.fr

July 12, 2006

## Abstract

We prove that the codes issued from the elimination of any sub-alphabet in a trace monoid are finite-state recognizable. This implies in particular that the transitive factorizations of the trace monoids are recognizable by (boolean) finite-state automata.

**Keywords:** Trace monoid; Lazard's elimination; automata with multiplicities.

## 1 Introduction

Sch utzenberger ([8] Chapter 5) introduced the notion of a factorization of a monoid  $M$

$$M = \prod_{i \in I}^{\rightarrow} M_i \quad (1)$$

where  $(M_i)_{i \in I}$  is a subfamily of submonoids of the given monoid  $M$ . When  $M = A^*$  is a free monoid, at the both ends of the chain, one has complete factorizations like Lyndon and Hall factorizations [11] and the bisections  $|I| = 2$  [7].

A nice way to produce factorizations is to start with a bisection  $M = M_1 M_2$  and refine the factors using a uniform process. Doing this, we could obtain a complete factorization for every trace monoid [3]. Trace monoids are defined as follows. Consider an alphabet  $\Sigma = \{x_1, \dots, x_n\}$  and a commutation relation  $\vartheta$  (*i.e.* a reflexive and symmetric relation) on  $\Sigma$ . The trace monoid  $\mathbf{M}(\Sigma, \vartheta)$  is the quotient

$$\mathbf{M}(\Sigma, \vartheta) = \Sigma^* / \equiv_{\vartheta} \quad (2)$$

where  $\equiv_{\vartheta}$  is the congruence generated by the relators  $ab \equiv ba$  where  $(a, b) \in \vartheta$ .

Later on, we adressed the question of bisecting a trace monoid so that the left factor be generated by a subalphabet (Lazard bisection) and the right factor be a trace monoid [5]. Doing so, we obtained a complete description of the factors and graph-theoretical criteria for the factorization. We conjectured that the trace codes so obtained could be recognized by finite-state automata [5].

In this paper, we prove that the answer to the conjecture is positive. This will be a consequence of the more general result that if a trace monoid  $M(\Sigma, \vartheta)$  is bisected as

$$M(\Sigma, \vartheta) = L.M(B, \vartheta_B) \quad (3)$$

with  $B \subset \Sigma$  and  $\vartheta_B = \vartheta \cap (B \times B)$ , then the minimal generating set  $\beta(L)$  of  $L$  is recognizable by a finite-state, effectively constructible automaton. Here, we prove this fact and give the construction of the automaton.

The paper is organised as follows:

In section 2, we recall basic notions related to trace monoids and recognizability. In section 3, we prove that the left factor of a Lazard bisection is a recognizable set and we describe the construction of a deterministic automaton recognizing it in section 4. To end with, we explain in section 5 how to construct a deterministic automaton which recognizes the generating set of the left factor of such a bisection.

## 2 Trace Monoids

Trace monoids were introduced by Cartier and Foata with the purpose of studying some combinatorial problems linked with rearrangements (see [2]).

Next, this notion has been studied by Mazurkiewicz and many schools of Computer Sciences in the context of concurrent program schemes (see [9, 10]).

Let  $x \in \Sigma$  be a letter and denote  $\text{Com}(x)$  the set of letters which commute with  $x$

$$\text{Com}(x) = \{z \mid (x, z) \in \vartheta\}. \quad (4)$$

In particular, one has  $x \in \text{Com}(x)$ . Let  $w \in \mathbf{M}(\Sigma, \vartheta)$  be a trace, we will denote

$$TA(w) = \{x \in \Sigma \mid w = ux, u \in \mathbf{M}(\Sigma, \vartheta)\} \quad (5)$$

the *terminal alphabet* of  $w$ .

As it is shown in [3], Lazard elimination occurs in the context of traces. Let  $B$  be a subalphabet of  $\Sigma$  and  $\vartheta_B = \vartheta \cap (B \times B)$ . The trace monoid splits into two submonoids

$$\mathbf{M}(\Sigma, \vartheta) = L.\mathbf{M}(B, \vartheta_B) \quad (6)$$

where  $L$  is the submonoid consisting in the traces whose terminal alphabet is a subset of  $\Sigma \setminus B$ . Furthermore the decomposition is unique, which suggests that the following equality occurs in  $\mathbf{Z}\langle \Sigma, \vartheta \rangle = \mathbf{Z}[\mathbf{M}(\Sigma, \vartheta)]$ , the algebra of series corresponding to  $\mathbf{Z}[\mathbf{M}(\Sigma, \vartheta)]$  [4]. Thus,

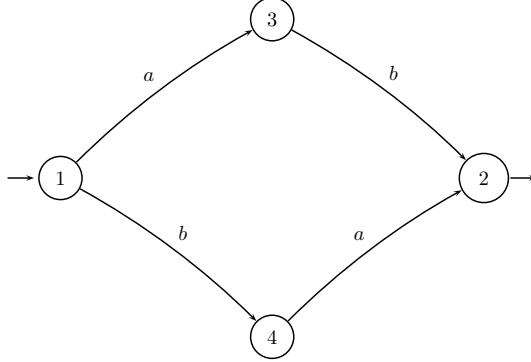
$$\underline{\mathbf{M}(\Sigma, \vartheta)} = \underline{L}.\underline{\mathbf{M}(B, \vartheta_B)} \quad (7)$$

where  $\underline{S}$  denotes the characteristic series of a subset  $S \subset \mathbf{M}(\Sigma, \vartheta)$  *i.e.*

$$\underline{S} = \sum_{w \in S} w \in \mathbf{Z}\langle \langle \Sigma, \vartheta \rangle \rangle. \quad (8)$$

Let  $\phi$  be the natural surjection  $\Sigma^* \rightarrow \mathbf{M}(\Sigma, \vartheta)$ , the *set of the representative words* of a trace  $t$  is defined as  $\text{Rep}(t) = \phi^{-1}(t)$ . We can extend this definition to trace langages  $\text{Rep}(L) = \phi^{-1}(L) = \bigcup_{t \in L} \phi^{-1}(t)$ . A trace langage is said *recognizable* if and only if its representative set is, and we say that an automaton recognizes  $L$  if and only if it *recognizes*  $\text{Rep}(L)$ .

**Example 1** Let  $a$  and  $b$  be two commuting letters, then the set  $\text{Rep}(\{ab\})$  is recognized by the automaton



In fact, one can prove that a rational language is a set of representatives (i.e. it is saturated w.r.t. the congruence  $\equiv_\theta$ ) if and only if the corresponding minimal automaton shows complete squares as above.

We will denote  $\text{Rec}(\Sigma, \vartheta)$  the set of recognizable sets of traces.

### 3 Recognizing the left factor

The  $\mathbf{Z}$ -rationality of the left factor  $L$  is a direct consequence of the unicity of the decomposition, which, in term of formal series, reads

$$\underline{\mathbf{M}}(\Sigma, \vartheta) = \underline{L} \cdot \underline{\mathbf{M}}(B, \vartheta_B). \quad (9)$$

where  $\underline{S}$  denotes the  $\mathbf{Z}$ -characteristic series of the set  $S$  (i.e.  $\underline{S} = \sum_{x \in S} x$ ). Indeed, by a classical result due to Cartier and Foata ([2] Theorem 2.4) the  $\mathbf{Z}$ -characteristic series of  $\underline{\mathbf{M}}(\Sigma, \vartheta)$  is rational when the alphabet  $\Sigma$  is finite<sup>a</sup>:

$$\underline{\mathbf{M}}(\Sigma, \vartheta) = \frac{1}{\sum_{\{a_1, \dots, a_n\} \in \mathbf{Cliques}(\Sigma)} (-1)^n a_1 \cdots a_n}. \quad (10)$$

where the sum at the denominator is taken over the set  $\mathbf{Cliques}(\Sigma)$  of the cliques of  $\Sigma$  (i.e. commutative sub-alphabets). Hence, one obtains the

<sup>a</sup>The formula holds also when the alphabet is infinite but the denominator is then a series.

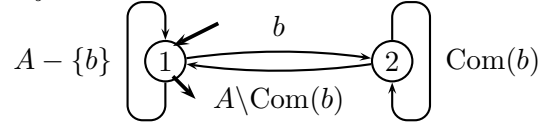
rational equality

$$\underline{L} = \frac{1}{\sum_{\{a_1, \dots, a_n\} \in \mathbf{Cliques}(\Sigma)} (-1)^n a_1 \cdots a_n} \times \left( \sum_{\{b_1, \dots, b_n\} \in \mathbf{Cliques}(B)} (-1)^n b_1 \cdots b_n \right) \quad (11)$$

Nevertheless, this remark is not sufficient to show that  $L$  is recognizable as a language. Furthermore, for traces, one has the strict inclusion  $\text{Rec}(\Sigma, \vartheta) \subset \text{Rat}(\Sigma, \vartheta)$ . To prove that  $L$  is recognizable it suffices to find a construction of  $\text{Rep}(L)$  using only recognizable operations. For each letter  $x \in \Sigma$ , let  $\mathbf{TN}_x$  be the set of representative words of traces whose terminal alphabet does not contain  $x$ . Remarking that  $\text{Rep}(L)$  is the representative set of the traces whose terminal alphabet contains no letter of  $B$ , one has

$$\text{Rep}(L) = \bigcap_{b \in B} \mathbf{TN}_b. \quad (12)$$

Hence,  $\text{Rep}(L)$  is recognizable if each  $\mathbf{TN}_b$  is. But, one can easily verify that automaton  $\mathcal{A}_b$ :



recognizes  $\mathbf{TN}_b$ . Thus, we have the proposition

**Proposition 1**  $L$  is a recognizable submonoid of  $M(\Sigma, \vartheta)$ .

## 4 A deterministic automaton for a terminal condition

One can compute a deterministic automaton recognizing  $L$  generalizing the construction of  $\mathcal{A}_b$ . We consider an automaton  $\mathcal{A}_B = (S_B, I_B, F_B, T_B)$  such that:

1. The set  $S_B$  of its states is the set of all the sub-alphabets of  $B$ ,
2. There a unique initial state  $I_B = \{\emptyset\}$ ,
3. There a unique final state  $F_B = \{\emptyset\} = I_B$ ,
4. The transitions are

$$T_B = \{(B', x, ((B' \cup \{x\}) \cap \text{Com}(x) \cap B))\}_{B' \subset B, x \in \Sigma}.$$

One has

**Proposition 2** *The automaton  $\mathcal{A}_B$  is a complete deterministic automaton recognizing  $\text{Rep}(L)$ .*

**Proof** It is straightforward to see that such an automaton is complete and deterministic. Now, let us prove that it recognizes  $\text{Rep}(L)$ . As  $\mathcal{A}_L$  is complete deterministic, for each word  $w = a_1 \cdots a_n$  we can consider a state  $s_w$  which is the state of  $\mathcal{A}_B$  after reading  $w$ . More precisely, we can define  $s_w$  as  $s_w = s_n$  in the following chain of transitions

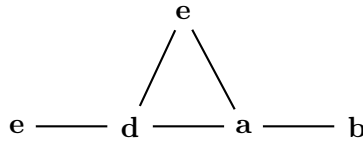
$$(\emptyset, a_1, s_1), (s_1, a_2, s_2), \dots, (s_{n-1}, a_n, s_n = s_w). \quad (13)$$

We first prove that if  $w$  is a word then  $s_w = TA(t_w) \cap B$  where  $t_w$  denotes the trace admitting  $w$  as representative word. We use an induction process, considering as starting point:  $(\emptyset, x, \{x\} \cap B)$  where  $x \in \Sigma$ . Let  $w = a_1 \cdots a_n$  be a word of length  $n$ , such that  $s_w$  is the intersection between  $B$  and the terminal alphabet of trace  $t_w$ . Let  $a_{n+1} \in \sigma$  be an other letter. One has,  $(s_w, a_{n+1}, s_{wa_{n+1}}) \in T_B$ . Hence, the set  $s_{wa_{n+1}}$  is

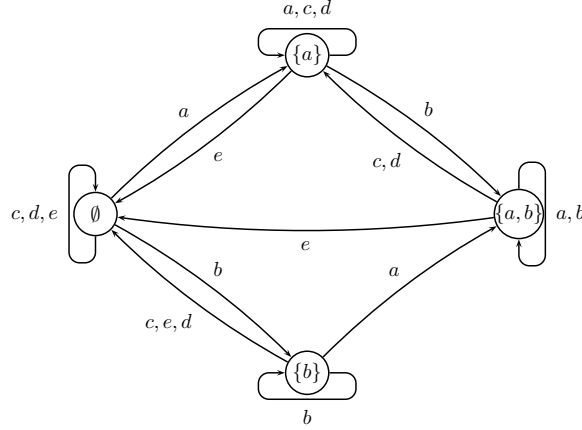
$$s_{wa_{n+1}} = (s_w \cup \{a_{n+1}\}) \cap \text{Com}\{a_{n+1}\} \cap B = TA(t_w a_{n+1}) \cap B = TA(t_{wa_{n+1}}) \cap B. \quad (14)$$

This proves our assertion. Then, the set of words  $w$  such that  $s_w = \emptyset$  is exactly the set of representative words of  $L$ .  $\square$

**Example 2** We consider the trace alphabet given by the following commutation graph



If we set  $B = \{a, b\}$ , then  $L$  is recognized by the following automaton (in the figure the only initial state and the only final state is  $\emptyset$ ):



## 5 A deterministic automaton for the generating set of the left factor

Each submonoid  $M$  of a trace monoid has an unique *generating set* which is the subset  $G(M) = M \setminus M^2$ .<sup>b</sup>

In this section, we prove that  $G(L)$  is recognizable and we construct an automaton  $A_\beta$  which recognizes it. The automaton  $A_\beta$  is obtained from  $A_B$  by adding two states  $F, H$ , choosing  $F$  as final state instead of  $\emptyset$  and modifying the transitions in such a way that if a letter of  $Z = A - B$  is read, the state reached belongs in  $F, H$  and the other states become unreachable.

More precisely, one considers the automaton  $\mathcal{A}_\beta = (S_\beta, I_\beta, F_\beta, T_\beta)$  obtained from the automaton  $A_B = (S_B, I_B, F_B, T_B)$  computed in the previous section as follows:

1. The set of its states  $S_\beta$ , is the set of the sub-alphabets of  $B$  plus two states  $F$  and  $H$ ,
2. There is a unique initial state  $I_\beta = \{\emptyset\}$ ,
3. There is a unique final state  $F_\beta = \{F\}$ ,
4. The transitions are

$$T_\beta = T_{B \rightarrow B} \cup T_{B \rightarrow F} \cup T_{B \rightarrow H} \cup T_{F \rightarrow H} \cup T_{H \rightarrow H}$$

<sup>b</sup>The fact that  $G(M)$  generates  $M$  is straightforward and the unicity comes from that the  $\mathbf{Z}$ -characteristic series of  $G(M)$  is the inverse of the  $\mathbf{Z}$ -characteristic series of  $M$  in  $\mathbf{Z}\langle\langle A \rangle\rangle$ .



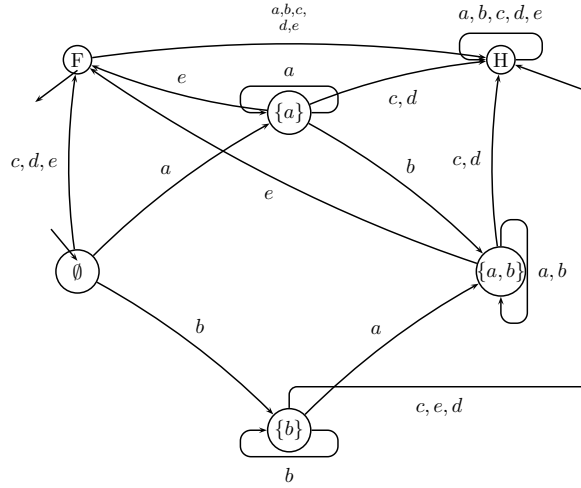
where

- (a)  $T_{B \rightarrow B} = \{(B', b, B'')\}_{B', B'' \subset B, b \in B, (B', b, B'') \in T_B}$ ,
- (b)  $T_{B \rightarrow F} = \{(B', z, F)\}_{(B', z, \emptyset) \in T_B, B' \subset B, B' \neq \emptyset, z \in Z}$ ,
- (c)  $T_{B \rightarrow H} = \{(B', z, H)\}_{(B', z, B'') \in T_B, B', B'' \notin \{\emptyset, F, H\}, z \in Z}$ ,
- (d)  $T_{F \rightarrow H} = \{(F, x, H)\}_{x \in \Sigma}$ ,
- (e) and  $T_{H \rightarrow H} = \{(H, x, H)\}_{x \in \Sigma}$ .

**Proposition 3** *The automaton  $\mathcal{A}_\beta$  recognizes  $\text{Rep}(G(L))$ .*

**Proof** The automaton is almost the same as  $\mathcal{A}_B$ . As for  $\mathcal{A}_B$ , if a word of  $B^*$  is read, the automaton is in the state corresponding to its terminal alphabet. The difference appears when a letter of  $Z$  is read, if it is read from the  $\emptyset$  state the automaton goes to the state  $F$ . Consider now a word  $w = w'z$  with  $w' \in B^+$ ,  $z \in Z$ . We denote  $\delta_w$  the state of the automaton after reading  $w$  (this definition makes sense as, like  $\mathcal{A}_B$ ,  $\mathcal{A}_\beta$  is deterministic). Now, if  $\{z\} = TA(w'z)$ , then  $(\delta'_w, z, F) \in T_{B \rightarrow F}$  which means that  $w$  is recognized by  $\mathcal{A}_\beta$ , otherwise  $(\delta'_w, z, H) \in T_{B \rightarrow H}$  and  $w$  is not recognized by  $\mathcal{A}_\beta$ . Furthermore, for each  $z \in Z$  and  $b \in B$ ,  $\delta_{w'zaw''} = H$  (for each  $w', w'' \in \Sigma^*$ ). This ends the proof.  $\square$

**Example 3** Consider again the example (2). Then,  $\beta$  is recognized by the automaton



## 6 Conclusion

The factorisations of free monoids (or in a more general setting of a monoid constructed by generators and relations) is a relevant topic in the context of the theory of codes [1]. Lazard bisections, or more generally rational bisections [7], play a role in the construction of bases of free Lie algebras [11] and the study of circular codes [1, 11]. A natural question asks if it is possible to generalize these properties to other monoids in particular when the free module over these monoids can be endowed with a shuffle coproduct [6]. The results contained in the paper consist in a step in the study of these problems for the trace monoids. The role played by the Lazard bisections in this context is not still completely known (see [3, 5] for some results).

## References

- [1] J. Berstel and D. Perrin , Theory of codes (Pure and applied Mathematics Acad. Press. Inc., 1995).
- [2] P. Cartier and D. Foata, *Problèmes combinatoires de commutation et réarrangements*, Lect. Not. In Math., **85** (1969).
- [3] G. Duchamp and D. Krob , *Factorisations dans le monoïde partiellement Commutatif libre*, C.R. Acad. Sci. Paris, **312**, série I (1991), 189-192.
- [4] Duchamp G., Krob D., *Combinatorics on traces*, Ch II du "Book of traces" (Ed. G. Rozenberg, V. Dieckert), World Scientific (1995).
- [5] G. Duchamp and J.-G. Luque, *Transitive factorisations of partially Commutative free monoids and Lie algebras.*, Discrete Math., **246** (2002).
- [6] G. Duchamp and J.G. Luque, *Congruences compatible with the shuffle product*, proceeding of Formal Power series and algebraic combinatorics FPSAC'00, Moscow, 2000
- [7] G. Duchamp and J.-Y. Thibon, *Bisections reconnaissables*, Theoretical Informatics and Applications, **22** (1988), 113-128, (Gauthiers-Villars).
- [8] N. Lothaire, *Combinatorics on words*, Cambridge University Press (1997).

- [9] A. Mazurkiewicz, *Concurrent program schemes and their interpretation*, (DAIMI Rep., PB 78, Aarhus University, 1977).
- [10] A. Mazurkiewicz, *Traces, histories and graph: instances of process monoid*, Lect. Not. in Comp. Sci., **176**, 115-133 (1984).
- [11] C. Reutenauer, *Free Lie Algebras* (Oxford University Press, New-York, 1993)
- [12] G. Rozenberg, V. Dieckert, *The Book of traces* (World Scientific ,1995).
- [13] G.X. Viennot. *Heaps of pieces*, Lect. Notes in Math., **1234** (1986), 321-350.