Laboratoire d'Informatique de Paris Nord Université Paris 13 (LIPN UMR 7030 du CNRS), Avenue J.-B. Clément 93430 Villetaneuse, France Gerard.Duchamp@lipn.univ-paris13.fr Laboratoire d'informatique de l'Institut Gaspard Monge (UMR 8049 IGM-LabInfo), Université de Marne-la-Vallée, 5, Boulevard Descartes, 775454 Champs-sur-Marne, France Jean-Gabriel.Luque@univ-mlv.fr

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

Jean-Gabriel LUQUE

December 9, 2004

Abstract

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

Trace monoid; Lazard's elimination; automata with multiplicities.

1 Introduction

Schützenberger ([7] Chapter 5) introduced the notion of a factorization of a monoid M

$$M = \prod_{i\in I}^{\rightarrow} M_i \tag{1}$$

where $(M_i)_{i\in I}$ is a subfamily of submonoids of the given monoid M. At the two ends of the chain, one has complete factorizations like Lyndon and Hall factorizations [10] and the bisections $|I| = 2$ [6].
A nice way to produce factorizations is to start with a bisection $M = M_1M_2$ and refine the factors using a uniform process. Doing this, we could obtain a complete factorization for every trace monoid [3].
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 [4]. 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 [4].

In this paper, we prove that the answer of 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

$$M(\Sigma, \vartheta) = L.M(B, \vartheta_B) \tag{2}$$

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 traces 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 [8, 9]).

Consider an alphabet $\Sigma = \{x_1, \cdots, x_n\}$ and a commutation relation ϑ (*i.e.* a reflexive and symmetric relation) on Σ. The trace monoid $\mathbf{M}(\Sigma, \vartheta)$ is the quotient

$$\mathbf{M}(\Sigma, \vartheta) = \Sigma^* /_{\equiv_\vartheta} \tag{3}$$

where $\equiv_\vartheta$ is the congruence generated by the relators $ab \equiv ba$ where $(a, b) \in \vartheta$. Let $x \in \Sigma$ be a letter and denote $\mathrm{Com}(x)$ the set of letters which commute with x

$$\mathrm{Com}(x) = \{z | (x, z) \in \vartheta\}. \tag{4}$$

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

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

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 Σ and $\vartheta_B = \vartheta \cap (B \times B)$. The trace monoid splits into two submonoids

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

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 $\mathbf{Z}\langle \Sigma, \vartheta \rangle = \mathbf{Z}[\mathbf{M}(\Sigma, \vartheta)]$. Thus,

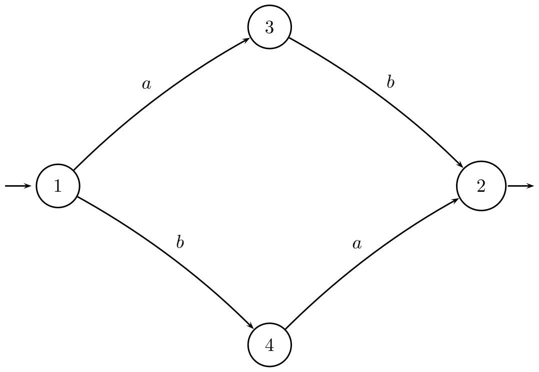
$$\underline{\mathbf{M}(\Sigma, \vartheta)} = \underline{L}.\underline{\mathbf{M}(B, \vartheta_B)} \tag{7}$$

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

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

Let ϕ be the natural surjection $\Sigma^* \to \mathbf{M}(\Sigma, \vartheta)$, the *set of the representative words* of a trace t is defined as $\mathrm{Rep}(t) = \phi^{-1}(t)$. We can extend this definition to trace langages $\mathrm{Rep}(L) = \phi^{-1}(L) = \bigcup_{t \in L} \phi^{-1}(t)$. 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* $\mathrm{Rep}(L)$.

Example 1 Let a and b be two commuting letters, then the set $\{ab\}$ 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 $\mathrm{Rec}(\Sigma, \vartheta)$ 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

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

where $\underline{S}$ denotes the $\mathbf{Z}$-characteristic series of the set S (*i.e.* $\underline{S} = \sum_{x \in S} x$). Indeed, when the alphabet Σ is finite[a] the $\mathbf{Z}$-characteristic series of $\underline{\mathbf{M}(\Sigma, \vartheta)}$ is rational:

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

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

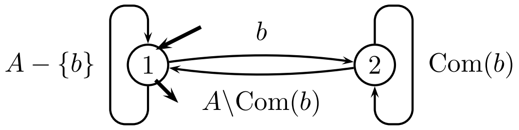
$$\underline{L} = \frac{1}{\displaystyle\sum_{\{a_1, \ldots, a_n\} \in \mathbf{Cliques}(\Sigma)} (-1)^n a_1 \cdots a_n} \times \left(\sum_{\{b_1, \ldots, b_n\} \in \mathbf{Cliques}(B)} (-1)^n b_1 \cdots b_n \right) \tag{11}$$

[a]The formula holds also when the alphabet is infinite but the denominator is then a series.

Nevertheless, this remark is not sufficient to show that L is recognizable as a language. Furthermore, for traces, one has the strict inclusion $\mathrm{Rec}(\Sigma, \vartheta) \subset \mathrm{Rat}(\Sigma, \vartheta)$. To prove that L is recognizable it suffices to find a construction of $\mathrm{Rep}(L)$ using only recognizable operations (like concatenation, union, intersection, complement...). 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 $\mathrm{Rep}(L)$ is the representative set of the traces whose terminal alphabet contains no letter of B, one has

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

Hence, $\mathrm{Rep}(L)$ 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

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

One has

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

It is straightforward to see that such an automaton is complete and deterministic. Now, let us prove that it recognizes $\mathrm{Rep}(L)$. 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

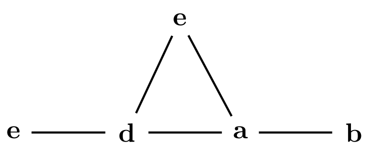
$$(\emptyset, a_1, s_1), (s_1, a_2, s_2), \cdots, (s_{n-1}, a_n, s_n = s_w). \tag{13}$$

We first prove that if w is a word then $s_w = AT(t_w) \cap B$ 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

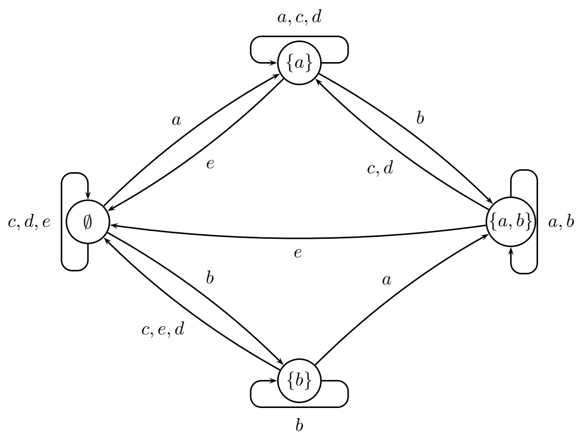
$$s_{wa_{n+1}} = (s_w \cup \{a_n\}) \cap \mathrm{Com}\{a_n\} \cap B = TA(ta_{n+1}) \cap B. \tag{14}$$

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

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

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

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

where

(a) $T_{B\to B} = \{(B', b, B'')\}_{B',B''\subset B, b\in B, (B',b,B'')\in T_B}$,

(b) $T_{B\to F} = \{(B', z, F)\}_{(B',z,\emptyset)\in T_B, B'\subset B, B'\neq\emptyset, z\in Z}$,

(c) $T_{B\to H} = \{(B', z, H)\}_{(B',z,B'')\in T_B, B',B''\not\in\{\emptyset,F,H\}, z\in Z}$,

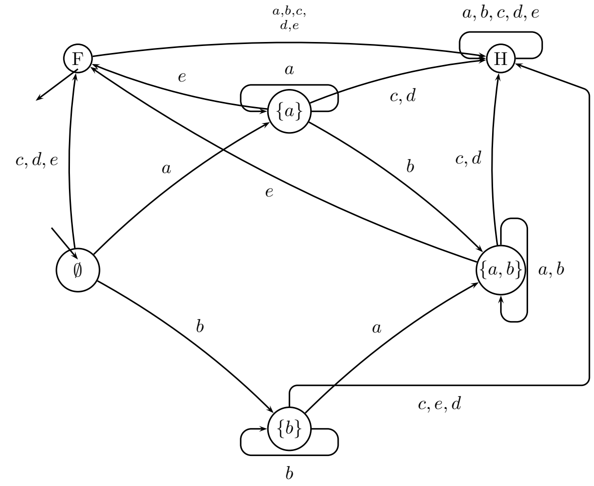
(d) $T_{F\to H} = \{(F, x, H)\}_{x\in\Sigma}$,

(e) and $T_{H\to H} = \{(H, x, H)\}_{x\in\Sigma}$.

Proposition 3 *The automaton $\mathcal{A}_\beta$ recognizes* $\mathrm{Rep}(\beta)$*, $\beta = L^*\backslash L^+$ denoting the generating set of the monoid L.*

The automaton is almost the same than $\mathcal{A}_B$. 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 δ_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 $(\delta'_w, z, F) \in T_{B\to F}$ which means that w is recognized by $\mathcal{A}_\beta$, otherwise $(\delta'_w, z, H) \in T_{B\to H}$ 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.

Example 3 Consider again the example (4). Then, β 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 [6], play a role in the construction of basis of free Lie algebras [10] and the study of circular codes [1, 10]. 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 [5]. 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, 4] 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] G. Duchamp and J.-G. Luque, *Transitive factorisations of partially Commutative free monoids and Lie algebras.*, Discrete Math., **246** (2002).

[5] G. Duchamp and J.G. Luque, *Congruences compatible with the shuffle product*, proceeding of Formal Power series and algebraic combinatorics FPSAC'00, Moscow, 2000

[6] G. Duchamp and J.-Y. Thibon, *Bisections reconnaissables*, Theoretical Informatics and Applications, **22** (1988), 113-128, (Gauthiers-Villars).

[7] N. Lothaire, Combinatorics on words, Cambridge University Press (1997).

[8] A. Mazurkievicz, *Concurrent program schemes and their interpretation*, (DAIMI Rep., PB 78, Aarhus University, 1977).

[9] A. Mazurckievicz, *Traces, histories and graph: instances of process monoid*, Lect. Not. in Comp. Sci., **176**, 115-133 (1984).

[10] C. Reutenauer, Free Lie Algebra (Oxford University Press, New-York, 1993)

[11] G. Rozenberg, V. Dieckert, *Combinatorics on traces*, The Book of traces (World Scientific ,1995).

[12] G.X. Viennot. *Heaps of pieces*, Lect. Notes in Math., **1234** (1986), 321-350.