

M. CLERBOUT

Y. ROOS

I. RYL

## **Semi-commutations and partial commutations**

*Informatique théorique et applications*, tome 34, n° 4 (2000),  
p. 307-330

[http://www.numdam.org/item?id=ITA\\_2000\\_\\_34\\_4\\_307\\_0](http://www.numdam.org/item?id=ITA_2000__34_4_307_0)

© AFCET, 2000, tous droits réservés.

L'accès aux archives de la revue « Informatique théorique et applications » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

## SEMI-COMMUTATIONS AND PARTIAL COMMUTATIONS

M. CLERBOUT<sup>1</sup>, Y. ROOS<sup>1</sup> AND I. RYL<sup>1</sup>

**Abstract.** The aim of this paper is to show that a semi-commutation function can be expressed as the compound of a sequential transformation, a partial commutation function, and the reverse transformation. Moreover, we give a necessary and sufficient condition for the image of a regular language to be computed by the compound of two sequential functions and a partial commutation function.

**AMS Subject Classification.** 68Q45, 68Q85.

### 1. INTRODUCTION

*Semi-commutations*, introduced in [2], are natural extensions of *partial commutations*, proposed by Mazurkiewicz [13] as formal tools for the modeling of concurrent systems. The bases of the theory of partial commutations stand at the meeting point of both *Petri nets* theory and *formal languages and automata* theory. A relation of partial commutation, so called relation of *independence*, is a symmetric and irreflexive binary relation defined on a finite alphabet. Letters represent actions and two letters are independent if the actions they represent are concurrent. This very simple definition has led to a very fruitful theory: the *theory of traces*, these last ones being the equivalence classes of the congruence generated on words by a relation of partial commutation. Among numerous and important results having contributed to develop this theory, let us quote the first-rate theorem of Zielonka [17] which, by defining a particular type of automaton, allows to define a strong notion of reconnaissability in traces; let us also quote the very nice theorem of Ochmański [15], generalizing the Kleene's theorem by introducing rational operations adapted to traces. The reader can refer to the state of the art of this theory in [7].

---

<sup>1</sup> LIFL, bâtiment M3, Cité Scientifique, 59655 Villeneuve-d'Ascq Cedex, France;  
e-mail: {clerbout,yroos,ryl}@lifl.fr

The only difference between partial commutations and semi-commutations is the relation of independence: it is not necessarily symmetric anymore. Therefore, semi-commutations are well adapted to the expression of synchronization between process like Producer/Consumer's problems or Readers/Editors' problems. For example, let us consider the alphabet  $\{c, p\}$  where  $c$  represents the *to consume* action and  $p$  represents the *to produce* action. Relation (or rule) of semi-commutation  $cp \rightarrow pc$  expresses the possibility to produce faster than to consume. So, the Dyck's language on the alphabet  $\{c, p\}$ , denoted by  $D_1^*(c, p)$ , and representing all the correct linkings of productions and consumptions in the case of a not bounded memory, is closed by rewriting using this rule of semi-commutation. More precisely,  $D_1^*(c, p)$  is the closure, under the semi-commutation function associated with the rule  $cp \rightarrow pc$ , of the language  $(pc)^*$  which represents a correct linking of productions and consumptions in the case of a buffer of size 1. As such, a semi-commutation function can be seen as a concurrency operator working on languages.

A natural question, when one introduces a new operation, is to try to express it by using more simple or already known operations. From this viewpoint, a decomposition theorem, expressed in [3], stipulates that any semi-commutation function is equal to the compound of a certain number of elementary semi-commutation functions named functions of atomic semi-commutations. Another result, expressed in [2] allows to express any commutation function as a composition of morphisms, inverse morphisms and partial commutations functions. The number of partial commutations functions occurring in this result is however very important.

In this article, we introduce, for any semi-commutation function  $f$ , a sequential transformation  $\tau_f$  such that  $f$  can be expressed as the compound of  $\tau_f$ , a function of partial commutation, and  $\tau_f^{-1}$ . This allows us to enunciate the main results of this work: we give a necessary and sufficient condition, concerning iterating factors of a language, so that the image of this language by a semi-commutation function can be expressed as the compound of two sequential rational functions and a partial commutation function.

This theorem evokes the results expressed by Arnold in [1] on *projective CCI sets of P-traces*, and also the type of constructions realized by Husson in [11] concerning **relèvements** towards the partially commutative monoids. It allows to throw a new light on a certain number of previous results which underscore differences between the partial commutations and the semi-commutations [6, 16].

## 2. PRELIMINARIES

### 2.1. NOTATIONS

In the following;  $\Sigma$  will denote a finite alphabet. We shall denote by  $\text{alph}(u)$  the alphabet of a word  $u$  and by  $\varepsilon$  the empty word.

We shall denote by  $\Pi_X$  the *projection* onto the alphabet  $X$ , *i.e.* the morphism defined by:

$$\begin{aligned} \Pi_X : \Sigma^* &\longrightarrow X^* \\ x &\longmapsto \begin{cases} x & \text{if } x \in X, \\ \varepsilon & \text{otherwise.} \end{cases} \end{aligned}$$

A word  $u$  is a *factor* of a word  $v$  if there exist two words  $w_1$  and  $w_2$  such that  $v = w_1uw_2$ . We denote by  $\text{RF}(w)$  (respectively  $\text{LF}(w)$ ) the set of *right factors* (respectively *left factors*) of the word  $w$ , that is:

$$\begin{aligned} \text{RF}(w) &= \{v \in \Sigma^* \mid \exists u \in \Sigma^*, w = uv\}, \\ \text{LF}(w) &= \{u \in \Sigma^* \mid \exists v \in \Sigma^*, w = uv\}. \end{aligned}$$

The set of iterating factors of a language  $L$  over  $\Sigma$  is:

$$\{u \in \Sigma^* \mid \exists v, w \in \Sigma^*, vu^*w \subseteq L\}.$$

A word  $u$  is said to be a subword of a word  $v$  if there exists words  $w_0, w'_1, w_1, \dots, w'_n, w_n$  such that  $v = w_0w'_1w_1 \dots w'_iw_i \dots w'_nw_n$  and  $u = w'_1w'_2 \dots w'_n$ .

For each word  $v$  of  $\Sigma^*$  and for each letter  $a$  of  $\Sigma$ ,  $v/a = v$  if  $a \notin \text{LF}(v)$  and  $v/a = a^{-1}v$  otherwise.

Let us consider a rewriting system  $R$ . We shall write  $u \longrightarrow_R v$  if there are a rule  $\alpha \longrightarrow \beta$  in  $R$  and two words  $w$  and  $w'$  such that  $u = w\alpha w'$  and  $v = w\beta w'$ .

We say that there exists a *derivation* from  $u$  to  $v$  denoted by  $u \longrightarrow_R^* v$  if there are words  $w_0, w_1, \dots, w_n$ , ( $n \geq 0$ ), such that  $w_0 = u, w_n = v$ , and for each  $i < n, w_i \longrightarrow_R w_{i+1}$ . The integer  $n$  is called *derivation length*. When we have  $u \longrightarrow_R^* v$  with a known derivation of length  $n$ , we shall also write:  $u \longrightarrow_R^n v$ .

Let  $E$  be a set of integers. Then,  $\bar{E}$  denotes  $\mathbb{N} \setminus E$ . The cardinality of the finite set  $E$  is denoted by  $\|E\|$ . For a word  $u$ ,  $\|u\|_{ab}$  denotes the number of occurrences of the factor  $ab$  in  $\Pi_{\{a,b\}}(u)$ .

## 2.2. PARTIAL COMMUTATIONS AND SEMI-COMMUTATIONS

Let us recall some definitions and some results about partial commutations and semi-commutations (for more details see [7]).

**Definition 2.1.** A *partial commutation relation* over an alphabet  $\Sigma$  is an irreflexive and symmetrical relation included in  $\Sigma \times \Sigma$ . A *semi-commutation relation* over an alphabet  $\Sigma$  is an irreflexive relation included in  $\Sigma \times \Sigma$ .

**Definition 2.2.** With a semi-commutation (or a partial commutation)  $\theta$ , we associate a rewriting system, defined as:  $\{xy \longrightarrow yx \mid (x, y) \in \theta\}$  (note that all the rules of the system are symmetrical when  $\theta$  is a partial commutation). To simplify the notations, we also denote the rewriting system by  $\theta$ .

**Definition 2.3.** For a semi-commutation (or a partial commutation)  $\theta$  defined over an alphabet  $\Sigma$ , we denote by  $f_\theta$  the associated semi-commutation (or partial commutation) function. This function is defined by:

$$\forall u \in \Sigma^*, f_\theta(u) = \left\{ v \mid u \xrightarrow{\theta^*} v \right\}.$$

This definition is extended to languages:

$$\forall L \subseteq \Sigma^*, f_\theta(L) = \bigcup_{u \in L} f_\theta(u).$$

**Definition 2.4.** The non-commutation relation associated with the commutation (resp. semi-commutation) relation  $\theta$  is:

$$\bar{\theta} = (\Sigma \times \Sigma) \setminus \theta.$$

With each semi-commutation (or partial commutation)  $\theta$  defined over an alphabet  $\Sigma$  is associated an oriented graph (or non-oriented): the non-commutation graph  $(\Sigma, \bar{\theta})$  where  $\Sigma$  is the node set and  $\bar{\theta} \setminus \{(x, x) \mid x \in \Sigma\}$  the edge set.

We say that there exists a path from  $a_0$  to  $a_n$  in the non-commutation graph  $(\Sigma, \bar{\theta})$  if there exist  $a_1, \dots, a_{n-1}$  such that for each  $0 \leq i < n$ ,  $(a_i, a_{i+1})$  belongs to  $\bar{\theta}$ . Then,  $a_0 a_1 \dots a_n$  is a word labelling a path in  $(\Sigma, \bar{\theta})$ .

**Definition 2.5.** Let  $\theta$  be a semi-commutation (or a partial commutation) defined over an alphabet  $\Sigma$  and  $u$  be a word of  $\Sigma^*$ . The word  $u$  is said to be strongly connected (respectively connected) if the graph  $(\text{alph}(u), \bar{\theta})$ , which is the restriction of the non commutation graph of  $\theta$  to the alphabet of  $u$ , is strongly connected (respectively connected).

**Definition 2.6.** Any maximal connected subgraph (resp. strongly connected) of a graph  $G$  is named a connected component (resp. strongly connected component) of  $G$ .

In some proofs, we will use the notion of minimal derivation which is linked to the notion of distance between words.

**Definition 2.7.** Let  $\Sigma$  be an alphabet. We denote  $\Sigma_{\text{num}} = \Sigma \times \mathbb{N}$ .

We define inductively the application  $\text{num}: \Sigma^* \rightarrow \Sigma_{\text{num}}^*$  by:

- $\text{num}(\varepsilon) = \varepsilon$ ;
- $\forall u \in \Sigma^*, \forall a \in \Sigma, \text{num}(ua) = \text{num}(u)(a, |ua|_a)$ .

The morphism  $\text{denum}: \Sigma_{\text{num}}^* \rightarrow \Sigma^*$  is defined by:

- $\forall x = (a, i) \in \Sigma_{\text{num}}, \text{denum}(x) = a$ .

Let  $u, v \in \Sigma^*$  be two commutatively equivalent words. The distance from  $u$  to  $v$ , denoted  $d(u, v)$  is equal to  $\text{Card}(\{(a, b) \in \Sigma_{\text{num}} \times \Sigma_{\text{num}} \mid \text{num}(u) = xaybz\} \setminus \{(a, b) \in \Sigma_{\text{num}} \times \Sigma_{\text{num}} \mid \text{num}(v) = xaybz\})$ .

Let us remark that for each word  $u$ , the word  $\text{num}(u)$  contains only one occurrence of each letter. It is clear that if  $v \in f_\theta(u)$  for a semi-commutation  $\theta$ , then  $d(u, v)$  is a bound for the lengths of derivations from  $u$  to  $v$ . Moreover, the following lemma shows that each step decreasing the distance is a “good” step:

**Lemma 2.8. (Distance lemma [7])** *Let  $\theta$  be a semi-commutation over an alphabet  $\Sigma$ ,  $u$  a word of  $\Sigma^*$  and  $v \in f_\theta(u)$ . If  $u \xrightarrow{\theta} w$  with  $d(w, v) < d(u, v)$  then,  $v \in f_\theta(w)$ .*

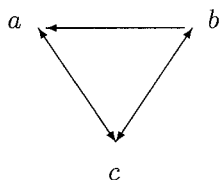
From this lemma follows the corollary saying that there always exists a minimal derivation of length  $d(u, v)$  between two words  $u$  and  $v$ .

**Corollary 2.9.** *Let  $\theta$  be a semi-commutation over an alphabet  $\Sigma$ ,  $u$  a word of  $\Sigma^*$  and  $v \in f_\theta(u)$ . There exists a derivation from  $u$  to  $v$  of length  $d(u, v)$ .*

### 3. IDEA OF THE CONSTRUCTION

Our aim is to show that it is possible to use a transformation, a partial commutation and a strictly alphabetical morphism to compute the closure under a semi-commutation of any language. In this section, we show on examples that the function can consist on adding different marks (colors or subscripts) to occurrences of letters. The marks distinguish occurrences of a same letter in order to allow them to commute – or not – depending on their position (relative to occurrences of other letters) in the starting word.

Let us consider a finite alphabet  $\Sigma = \{a, b, c\}$  and a semi-commutation  $\theta = \{(a, b)\}$  defined over  $\Sigma$ . The non-commutation graph of  $\theta$  is



The idea is to “color” the letters of a word  $u$  in such a way that two occurrences of letters  $a$  and  $b$  receive different colors when they commute in  $\Pi_{\{a,b\}}$  and that letters of the coloring of  $u$  having the same color are not allowed to commute.

To compute the closure under the semi-commutation  $\theta$  of a word  $u$ , we use a partial commutation and the colors that bring out some dependences of letters. As the only non-symmetrical rule is  $ab \rightarrow ba$ , we only have to color occurrences of letters  $a$  and  $b$ . Let us consider  $x$  and  $y$  two consecutive occurrences of letters in  $\Pi_{\{a,b\}}(u)$ . These  $x$  and  $y$  must receive distinct colors in the case when  $x = a$  and  $y = b$ . In all the other cases,  $x$  and  $y$  must receive the same color. We give to consecutive  $b$  and  $a$  or consecutive  $a$  or consecutive  $b$  the same color and we give to consecutive  $a$  and  $b$  distinct colors (when a  $b$  follows a  $a$ , we introduce a new color). Using this method, for a word  $u = cabbabbaaabab$  we obtain for example the colored word  $v = ca_0b_1b_1a_1b_2b_2a_2a_2a_2b_3a_3b_4$  (where colors are marked by

subscripts). We define now the partial commutation  $\varrho$  by:

$$\varrho = \{(a_i, b_j) \mid i \neq j\} \cup \{(b_i, a_j) \mid i \neq j\}.$$

As  $c$  does not belong to any couple of  $\theta$ , it does not receive any color and it does not belong to any couple of  $\varrho$ . Now, we denote by  $\varphi$  the strictly alphabetical morphism which removes colors and it is easy to see that  $f_\theta(u) = \varphi \circ f_\varrho(v)$ .

Clearly, this method may lead to use an infinite number of colors to compute the closure of an infinite language. We can notice that the number of colors may be infinite even when the beginning language is a regular language closed under  $\theta$  (consider for example the language  $(acbc)^*$ ).

In order to solve this problem, we can remark that, in some cases, it is possible to reuse several times a color in the same word. Consider for example the word  $u = abcaabacb$ . The first  $a$  receives the first color that is to say 0. The first  $b$  is allowed to commute with  $a$  so it receives another color, for example 1. The second and the third  $a$  receive the same color than the previous  $b$ : 1. For the moment we have  $a_0b_1ca_1a_1bacb$ . The following  $b$  may receive 2 but this  $b$  and the first  $a$  are the first and the last letter of a subword of  $u$  which is a word labelling a path of the non-commutation graph of  $\theta$  so, these two occurrences of letters will never be consecutive in any word derived from  $u$ . Thus, we can reuse the same color and mark this  $b$  with 0. For the same reason, the last  $b$  can also receive the color 0, we obtain the word  $v = a_0b_1ca_1a_1b_0a_0cb_0$ . We have again  $f_\theta(u) = \varphi \circ f_\varrho(v)$ .

With a partial commutation, we loose the non-symmetry. The use of the coloring makes up for this lost of power because the coloring contains some informations about dependence of actions.

We define in Section 4, a coloring using the same idea than in the previous example. We color words by adding to each letter a color for each sub-alphabet of two letters it belongs to. If a couple  $(a, b)$  belongs to  $\theta \cap \theta^{-1}$  or to  $\bar{\theta} \cap \bar{\theta}^{-1}$  then the colors of occurrences of letters  $a$  and  $b$  associated with  $\{a, b\}$  is always 0. We color a word from left to right and, for each  $(a, b)$  belonging to  $\theta \cap \bar{\theta}^{-1}$ , we color letters  $a$  and  $b$  in the following way:

- we start with color 0;
- an occurrence of letter  $a$  always receives the same color as the previous occurrence of letter of  $\{a, b\}$ ;
- an occurrence of letter  $b$  receives the least possible color. It can be an already used color if the occurrence of letter that has received this color is "separated" from this  $b$  by a strongly connected word (*i.e.* is the first letter of a factor whose last letter is this  $b$  and which contains as subword a strongly connected word starting with  $a$  and finishing with this  $b$ ). If it is not possible, the  $b$  receives the least color that has not already been used.

### 4. CONSTRUCTION

We have first to define the coloring we will use. We need a colored alphabet (here colors are just represented by counters) which only depends on the beginning alphabet.

**Definition 4.1.** *Let  $\Sigma$  be a finite alphabet. The colored alphabet corresponding to  $\Sigma$  is defined by:*

$$\Sigma_c = \{(x, c) \mid x \in \Sigma, c : \Sigma \longrightarrow \mathbb{N} \text{ such that } c(x) = 0\}.$$

The strictly alphabetical morphism used to remove colors only depends on the beginning alphabet:

**Definition 4.2.** *Let  $\Sigma$  be a finite alphabet. The strictly alphabetical morphism  $\varphi_\Sigma$  is defined by:*

$$\begin{aligned} \varphi_\Sigma : \quad \Sigma_c^* &\longrightarrow \Sigma^* \\ (a, c) &\longmapsto a. \end{aligned}$$

Now, we can propose a coloring which uses the idea explained in the previous section and which is, in general, defined as an infinite transducer. The definition of the transducer needs two preliminary definitions of sets of words depending of the non-commutation graph of the semi-commutation.

**Definition 4.3.** *Let  $\theta$  be a semi-commutation defined over an alphabet  $\Sigma$ . The set  $\mathcal{P}_{\theta, \Sigma}$  contains words labeling paths of the non-commutation graph of  $\theta$  and is defined by:*

$$\mathcal{P}_{(\Sigma, \theta)} = \{x_1 \dots x_n \mid (x_1, x_n) \in \theta \cap \bar{\theta}^{-1}, \forall i < j \leq n, (x_i \neq x_j \text{ and } (x_i, x_{i+1}) \in \bar{\theta})\}.$$

Note that the set  $\mathcal{P}_{(\Sigma, \theta)}$  is finite.

**Definition 4.4.** *Let  $\theta$  be a semi-commutation defined over an alphabet  $\Sigma$ . The set  $\mathcal{T}_{(\Sigma, \theta)}$  is defined by:*

$$\mathcal{T}_{(\Sigma, \theta)} = \{xy \mid (x, y) \in \theta \cap \bar{\theta}^{-1}, (\exists u \in \Sigma^*, xuy \in \mathcal{P}_{\theta, \Sigma})\}.$$

**Definition 4.5.** *Let  $\theta$  be a semi-commutation defined over an alphabet  $\Sigma$ . The transducer  $\tau_{(\Sigma, \theta)}$  is defined by*

$$\tau_{(\Sigma, \theta)} = (\Sigma, \Sigma_c, Q, \delta, q_0, Q)$$

where:

- $\Sigma$  is the input alphabet and  $\Sigma_c$  the output alphabet;
- $Q = 2^{(\Sigma \times \mathcal{P}_{(\Sigma, \theta)} \times \text{RF}(\mathcal{P}_{(\Sigma, \theta)}) \times 2^{\mathbb{N}}) \cup (\Sigma \times \mathcal{T}_{(\Sigma, \theta)} \times 2^{\mathbb{N}})}$  is the state set;
- $q_0 = \emptyset$  is the initial state;



- for each  $q$  of  $Q$  and each  $a$  of  $\Sigma$ , we have  $\delta(q, a) = (q', (a, c))$ . Let us set, for each  $q$  of  $Q$

$$E_{q,xy} = \bigcup_{(z,xyv,E) \in q, (z,xy,E) \in q} E,$$

then

–  $q'$  is defined by:

$$\begin{aligned} q' = & \{(a, ay, uy, \emptyset) \mid y \in \Sigma, ay \in \mathcal{P}_{(\Sigma, \theta)}\} \\ \cup & \{(z, xuy, v/a, E) \mid (z, xuy, v, E) \in q, x, y \in \Sigma, x \neq a, \\ & y \neq a\} \\ \cup & \{(a, ay, v, E) \mid (z, ay, v, E) \in q, y \in \Sigma\} \\ \cup & \{(a, xua, v, F) \mid ((z, xua, v, E) \in q), x \in \Sigma, v \neq a \wedge \\ & (\bar{A}(z', xu'a, a, E') \in q \text{ with } E' \subset E), \\ & F = E \text{ if } a = z \text{ and} \\ & F = E \cup \{\min(\bar{E}_{q,xa})\} \text{ otherwise}\} \\ \cup & \{(a, ay, \emptyset) \mid ay \in \mathcal{T}_{(\Sigma, \theta)} \text{ and } \bar{A}(z, ay, E) \in q\} \\ \cup & \{(z, xy, E) \mid (z, xy, E) \in q, x \neq a, y \neq a\} \\ \cup & \{(a, ay, E) \mid (z, ay, E) \in q\} \\ \cup & \{(a, xa, F) \mid (z, xa, E) \in q, \\ & F = E \text{ if } a = z \text{ and} \\ & F = E \cup \{\min(\bar{E}_{q,xa})\} \text{ otherwise}\}, \end{aligned}$$

– for each  $x$  of  $\Sigma$ ,  $c(x)$  is defined by  $c(x) = \min(\bar{E}_{q',xa} \cap \bar{E}_{q',ax})$ .

Words of the set  $\mathcal{P}_{(\Sigma, \theta)}$  are essential for the coloring, they are used to reduce the number of colors used for a given word. The informations contained in the states of the transducer are used to detect subwords of this set in words that have to be colored. We will speak about “rigid words”:

**Definition 4.6.** Let  $u$  be a word of  $\Sigma^*$ ,  $a$  and  $b$  be two letters of  $\Sigma$ . The word  $aub$  is rigid for  $ab$  if and only if there exists a subword  $u'$  of  $u$  such that  $au'b$  belongs to  $\mathcal{P}_{\theta, \Sigma}$ .

**Definition 4.7.** For a given letter  $(x, c)$  of a colored alphabet, the color associated with the sub-alphabet  $\{x, y\}$  is given by  $c(y)$ . By extension, in a word  $u = x_1x_2 \dots x_n$  where each  $x_j$  belongs to  $\Sigma$  the color associated with the sub-alphabet  $\{x, y\}$  of an occurrence  $x_i$  of a letter  $x$  is  $c(y)$  where:

- $\tau_{(\Sigma, \theta)}(u) = x'_1x'_2 \dots x'_n$  with for each  $1 \leq j \leq n$ ,  $x'_j \in \Sigma_c$ ;
- $x'_i = (x, c)$ .

Let us now present some terminologies and explain the way the transducer proceeds. For each  $(a, y)$  in  $\theta \cap \bar{\theta}^{-1}$ , and as soon as the first occurrence of  $a$  in a word is read, the transducer memorizes in its states the set of colors that cannot be the color associated with  $\{a, y\}$  of the next occurrence of  $y$ . The way to keep these informations in states varies from case to case:

- if  $ay$  is a word of  $\mathcal{T}_{(\Sigma, \theta)}$ , there does not exist any  $(z, auy, v, F)$  in any state of the transducer. The state  $q$  reached after the read of first occurrence of  $a$  of a word to be colored contains the tuple  $(a, ay, \emptyset)$ . This tuple evolves in each state accessible from  $q$  to take into account the colors associated with  $\{a, y\}$  of all occurrences of  $a$  and  $y$  that appear in the outputs on the path from  $q$  to the current state, moreover there is exactly one tuple of the form  $(z, ay, E)$  in each of these accessible states. So, for each state  $q'$  containing  $(z, ay, E)$ , we have  $E_{q', ay} = E$ ;
- if  $ay$  is not a word of  $\mathcal{T}_{(\Sigma, \theta)}$ , each time an occurrence of the letter  $a$  is read, we introduce, in the state reached with this reading, a tuple  $(a, auy, uy, \emptyset)$  for each word  $auy$  of  $\mathcal{P}_{(\Sigma, \theta)}$ : these tuples take into account the fact that this occurrence of  $a$  can be the first letter of a rigid factor of the word that is being read. These tuples are going to evolve in the next states to take into account the successive readings of letters of  $uy$  till they are removed during a transition. Note that only the reading of an occurrence of  $y$  may remove such a tuple from a state. A tuple  $(y, auy, v, E)$  is removed from a state  $q$  when  $(z, au'y, y, E)$  is in the state just before  $q$  with  $au'y$  a subword of  $auy$ : the last  $y$  may receive a color already used in the past.

Whatever are the further values of a tuple, we shall always refer to the occurrence of letter that initiates it. We also refer to the representations of a tuple. Let us formalize these notions:

**Definition 4.8.** *Let  $q$  be a state of the transducer  $\tau_{(\Sigma, \theta)}$  such that  $\delta(q_0, w) = (q, \tau_{(\Sigma, \theta)}(w))$  with  $w$  a word of  $\Sigma^*$  and  $t_q = (x, auy, v, E)$  (resp.  $t_q = (x, ay, E)$ ) be a tuple belonging to  $q$  with  $x, a, y$  letters of  $\Sigma$ . Then, the last occurrence of  $a$  in  $w$  such that the state reached after its reading contains  $(a, auy, uy, \emptyset)$  (resp.  $(a, ay, \emptyset)$ ) is said to be the occurrence that initiates  $t_q$ .*

**Definition 4.9.** *Let  $q$  be a state of the transducer  $\tau_{(\Sigma, \theta)}$  and  $t_q = (x, aub, v, E)$  (resp.  $t_q = (x, ab, E)$ ) be a tuple belonging to  $q$ . Let  $q'$  be such that  $\delta(q, y) = (q', (y, c))$ . Then the representations of  $t_q$  are defined as follow:*

- if  $y$  is different from  $a$  and  $b$ , then  $(x, aub, v/y, E)$  of  $q'$  is a representation of  $t_q$ ;
- if  $y$  is equal to  $a$  and  $E$  is not empty, then  $(a, aub, v, E)$  of  $q'$  is a representation of  $t_q$ ;
- if  $y$  is equal to  $b$ ,  $v \neq b$  and if there exists no  $(z, au'b, b, E')$  in  $q$  such that  $E' \subset E$ , then by definition of the transducer, there exists a tuple  $(b, aub, v, F)$  in  $q'$  with  $F$  equal to  $E$  when  $x = b$  and equal to  $E \cup \{\min(\bar{E}_{q, ab})\}$  otherwise. Then this tuple of  $q'$  is a representation of  $t_q$ ;
- all the representations of a representation of  $t_q$  are representations of  $t_q$ .

Let us remark that, from the definition of the transducer, whenever  $(x, a)$  belongs to  $\theta \cap \theta^{-1}$  or  $\bar{\theta} \cap \bar{\theta}^{-1}$ , for each state  $q$  accessible from  $q_0$ , the sets  $E_{q, xa}$  and  $E_{q, ax}$  are always empty, since neither  $xa$  belongs to  $\mathcal{T}_{(\Sigma, \theta)}$  nor there exists any word  $xua$  or  $aux$  in  $\mathcal{P}_{(\Sigma, \theta)}$ . So, in this case, in any word produced by the transducer, the color associated with  $\{x, a\}$  of each the occurrences of  $x$  and  $a$  is 0.

In the same way, if  $(x, a)$  belongs to  $\theta \cap \bar{\theta}^{-1}$ , for each state  $q$  accessible from  $q_0$ , the set  $E_{q,ax}$  is empty.

Conversely, if  $(x, a)$  belongs to  $\theta \cap \bar{\theta}^{-1}$ , (i.e. when the use of colors is essential) the transducer assures that only the reading of an  $a$  may affect the set of colors associated with  $\{x, a\}$  of following occurrences of  $x$  or  $a$ .

**Lemma 4.10.** *Let  $\theta$  be a semi-commutation over  $\Sigma$ ,  $(x, a)$  belong to  $\theta \cap \bar{\theta}^{-1}$ ,  $\delta(q_0, w) = (q, \tau_{(\Sigma, \theta)}(w))$  and  $\delta(q, t) = (q', (t, c))$  be transitions of  $\tau_{(\Sigma, \theta)}$  with  $w$  a word of  $\Sigma^*$  and  $t$  a letter of  $\Sigma$  different from  $a$ . Then, we have  $E_{q,xa} = E_{q',xa}$ .*

*Proof.* If  $xa$  belongs to  $\mathcal{T}_{(\Sigma, \theta)}$ , let us consider two cases. If  $t$  is different from  $x$ , then  $q$  contains a tuple  $(z, xa, E)$  if and only if  $q'$  contains  $(z, xa, E)$ , so  $E_{q,xa} = E = E_{q',xa}$ . If  $t$  is equal to  $x$  then either there exists a tuple  $(z, xa, E)$  in  $q$  so  $(x, xa, E)$  is in  $q'$  and  $E_{q,xa} = E = E_{q',xa}$ , or there is no tuple  $(z, xa, E)$  in  $q$  so  $(x, xa, \emptyset)$  is in  $q'$  and  $E_{q,xa} = E_{q',xa} = \emptyset$ .

When  $xa$  does not belong to  $\mathcal{T}_{(\Sigma, \theta)}$ , if  $t$  is different from  $x$ , clearly there exists  $(z, xua, v, E)$  in  $q$  if and only if  $q'$  contains a  $(z, xua, v', E)$  (with  $v' = v/t$ ). So, we have  $E_{q,xa} = \bigcup_{(z, xua, v, E) \in q} E = \bigcup_{(z, xua, v', E) \in q'} E = E_{q',xa}$ .

If  $t$  is equal to  $x$ , then each tuple  $(z, xua, v, E)$  in  $q$  is represented in  $q'$  by a tuple  $(x, xua, v, E)$ . Conversely, each tuple  $(z, xua, v, E)$  in  $q'$  is such that

- $z = x$ ;
- either  $E = \emptyset$  ( $x$  initiates the tuple  $(x, xua, v, \emptyset)$  in  $q'$ );
- or  $E \neq \emptyset$ , then this tuple is a representation of a tuple of  $q$  whose value is  $(z, xua, v, E)$ .

So, as previously, we have  $E_{q,xa} = \bigcup_{(z, xua, v, E) \in q} E = \bigcup_{(z, xua, v', E) \in q'} E = E_{q',xa}$ . □

We deduce directly from this lemma that when  $(x, a)$  belongs to  $\theta \cap \bar{\theta}^{-1}$  two occurrences of  $x$  in a word  $w$  that are not separated by any occurrence of an  $a$  in the projection of  $w$  on  $\{x, a\}$  will be colored in the same way relatively to  $\{x, a\}$ :

**Property 4.11.** *Let  $(x, a)$  belong to  $\theta \cap \bar{\theta}^{-1}$ ,  $w = w_1 x w_2 x w_3$  be a word of  $\Sigma^*$  with  $|w_2|_a = 0$ , and  $\tau_{(\Sigma, \theta)}(w) = \tau_{(\Sigma, \theta)}(w_1)(x, c)w'_2(x, c')w'_3$  with  $\varphi_\Sigma(w'_2) = w_2$ . Then, we have  $c(a) = c'(a)$ .*

The next unvarying property describes more formally the content of each state of the transducer.

**Lemma 4.12.** *Let  $w$  be a word of  $\Sigma^*$  and  $(q, \tau_{(\Sigma, \theta)}(w)) = \delta(q_0, w)$  be a calculus of  $\tau_{(\Sigma, \theta)}$ . Then, the following properties hold.*

- I1. *If  $(y, xuy, v, E)$  belongs to  $q$  then:*
  - (a) *the last occurrence of a letter of  $\{x, y\}$  in  $w$  is an  $y$ ;*
  - (b) *the color for  $\{x, y\}$  of the last  $y$  in  $w$  is  $\min(\bar{E}_{q,xy})$ ;*
  - (c)  *$E$  is a set containing the colors associated with  $\{x, y\}$  of all occurrences of  $x$  in  $w$  since the one that initiates this tuple.*
- I2. *If  $(x, xuy, v, E)$  belongs to  $q$  then:*
  - (a) *the last occurrence of a letter of  $\{x, y\}$  in  $w$  is an  $x$ ;*

- (b) the color associated with  $\{x, y\}$  of the last  $x$  in  $w$  is  $\min(\bar{E}_{q,xy})$ ;
- (c) if  $E = \emptyset$  then there is no occurrence of  $y$  in  $w$  after the occurrence of  $x$  that initiates this tuple;
- (d) if  $E \neq \emptyset$  then:
- (i) there is at least one occurrence of  $y$  in  $w$  after the occurrence of  $x$  that initiates the tuple;
  - (ii)  $E$  is a set containing the colors associated with  $\{x, y\}$  of all occurrences of  $x$  in  $w$  since the one that initiates this tuple, except the color of the last  $x$  in  $w$ .
13. If  $(x, xy, E)$  belongs to  $q$  then:
- (a) the last occurrence of a letter of  $\{x, y\}$  in  $w$  is an  $x$ ;
  - (b) the color associated with  $\{x, y\}$  of the last occurrence of  $x$  in the word  $w$  is  $\min(\bar{E}_{q,xy}) = \min(\bar{E})$ ;
  - (c)  $E$  is a set containing the colors associated with  $\{x, y\}$  of all the occurrences of  $x$  in  $w$  except the color of the last  $x$  of  $w$ .
14. If  $(y, xy, E)$  belongs to  $q$  then:
- (a) the last occurrence of a letter of  $\{x, y\}$  in  $w$  is an  $y$ ;
  - (b) the color associated with  $\{x, y\}$  of the last occurrence of  $y$  in the word  $w$  is  $\min(\bar{E}_{q,xy}) = \min(\bar{E})$ ;
  - (c)  $E$  is a set containing the colors associated with  $\{x, y\}$  of all the occurrences of  $x$  in  $w$ .

*Proof.* For each part of the invariant, properties (a) and (b) are directly deduced from the definition of the transducer. So, we only prove the other properties by induction on the length of  $w$ .

The empty word clearly verifies these properties.

Let  $w' = wa$ ,  $\delta(q_0, w) = (q, \tau_{(\Sigma, \theta)}(w))$  and  $\delta(q, a) = (q', (a, c))$ . From the definition of the transducer,  $q'$  contains exactly the following tuples:

- $(a, auy, uy, \emptyset)$  for each word  $auy$  in  $\mathcal{P}_{(\Sigma, \theta)}$  with  $y$  a letter of  $\Sigma$ . As the occurrence of  $a$  read during the transition from  $q$  to  $q'$  initiates the tuple, *I2.c* is clearly true;
- $(a, auy, v, E)$  for each tuple  $(z, auy, v, E) \in q$  with  $y$  a letter of  $\Sigma$ . If  $E$  is empty, *I2.c* is true in  $q$  and since the last letter of  $wa$  is an  $a$ , it is also true in  $q'$ . If  $E$  is not empty, by induction hypothesis, *I2.d.i* is true in  $q$ . Since the occurrence of  $a$  read between  $q$  and  $q'$  do not initiate the tuple, *I2.d.i* is still true in  $q'$ . Now we still have to show that *I2.d.ii* holds. From Lemma 4.10, we get  $\min(\bar{E}_{q',ay}) = \min(\bar{E}_{q,ay})$ , there are two cases:
  - $z = a$ , *I2.d.ii* holds in  $q$  and as  $c(y) = \min(\bar{E}_{q',ay}) = \min(\bar{E}_{q,ay})$ , *I2.d.ii* is also true in  $q'$ ;
  - $z = y$ , by induction, *I1.c* is true in  $q$  and since  $c(y) \notin E$  (we have  $c(y) = \min(\bar{E}_{q',ay}) = \min(\bar{E}_{q,xy})$ ), *I2.d.ii* holds in  $q'$ ;
- $(z, xuy, v/a, E)$  for each  $(z, xuy, v, E)$  of  $q$  with  $x$  and  $y$  letters of  $\Sigma$  different from  $a$ . As the read of the last  $a$  of  $wa$  may only modify the third component of such tuples, each property true in  $q$  for a tuple  $(z, xuy, v, E)$  is also true in  $q'$  for  $(z, xuy, v/a, E)$  in  $q'$ ;

- $(a, xua, v, F)$  such that  $(z, xua, v, E)$  belongs to  $q$  and there does not exist a tuple  $(z', xu'a, a, E')$  in  $q$  with  $E' \subset E$ . If  $z$  is equal to  $a$ , as in this case  $F = E$  and  $I1.c$  is true in  $q$ ,  $I1.c$  is true in  $q'$ . If  $z$  is equal to  $x$  then, by induction hypothesis,  $I2$  is true in  $q$ . There are two cases:
  - if  $E$  is empty, there is no occurrence of  $a$  in  $w$  after the occurrence of  $x$  that initiates the tuple  $(z, xua, v, E)$  of  $q$ . From Property 4.11, the color associated with  $\{x, a\}$  of this occurrence is the same as the color of the last  $x$  in  $w$ , that is  $\min(\bar{E}_{q,xa})$ . As  $F$  is equal to  $\{\min(\bar{E}_{q,xa})\}$ ,  $I1.c$  is true in  $q'$ ;
  - if  $E$  is not empty,  $E$  is the set of the colors associated with  $\{a, x\}$  of all the occurrences of  $x$  since the one that initiates the tuple  $(z, xua, v, E)$  except the last one which is  $\min(\bar{E}_{q,xa})$ . As  $F$  is equal to  $E \cup \{\min(\bar{E}_{q,xa})\}$ ,  $I1.c$  is true in  $q'$ .
- $(z, xy, E)$  for each tuple  $(z, xy, E)$  of  $q$ , with  $a$  different from  $x$  and from  $y$ . In this case, since the read of  $a$  does not affect such tuples, if  $I3$  (resp.  $I4$ ) is true in  $q$ , it is also true in  $q'$ ;
- $(a, ay, \emptyset)$  for each word  $ay$  of  $\mathcal{T}_{(\Sigma, \theta)}$  such that there does not exist any tuple  $(z, ay, E)$  in  $q$ . As  $a$  is the first occurrence of  $\{a, y\}$  in  $wa$  and since  $E$  is empty,  $I3.c$  is clearly true in  $q'$ ;
- $(a, ay, E)$  for each  $(z, ay, E) \in q$ . There are two cases:
  - if  $z$  is equal to  $a$  then  $I3$  is true in  $q$ . So the the last occurrence of a letter of  $\{a, y\}$  in  $w$  is an  $a$ . From Property 4.11, this occurrence receives the same color than the last  $a$  of  $wa$ , that is to say  $\min(\bar{E})$ , so  $I3.c$  is true in  $q'$ ;
  - if  $z$  is equal to  $y$ ,  $I4$  is true in  $q$  so  $E$  is the set of the colors associated with  $\{a, y\}$  of all the occurrences of  $a$  in  $w$ . As  $c(y) = \min(\bar{E})$  does not belong to  $E$ ,  $I3.c$  is true in  $q'$ ;
- $(a, xa, F)$  for each  $(z, xa, E) \in q$ . There are two cases:
  - if  $z$  is equal to  $a$  then  $F$  is equal to  $E$ . Since  $I4.c$  is true in  $q$  for  $w$ ,  $I4.c$  is also true in  $q'$  for  $wa$ ;
  - if  $z$  is equal to  $x$  then  $I3$  is true in  $q$  so  $E$  is the set of the colors associated with  $\{x, a\}$  of all the occurrences of  $x$  in  $w$  except the last one which is  $\min(\bar{E})$ . As  $F$  is equal to  $E \cup \min(\bar{E})$ ,  $I4.c$  is true in  $q'$  for  $wa$ .

□

As last property of the transducer, we show that the form of the tuples present in a state gives us some informations about the occurrences that initiates them.

**Lemma 4.13.** *Let  $w$  be a word of  $\Sigma^*$  and  $(q, \tau_{(\Sigma, \theta)}(w)) = \delta(q_0, w)$  be a calculus of  $\tau_{(\Sigma, \theta)}$ . If  $(a, xua, v, E)$  and  $(a, xu'a, v', E')$  belong to  $q$  with  $E' \subset E$  then the occurrence of  $x$  that initiates  $(a, xua, v, E)$  precedes in  $w$  the one that initiates  $(a, xu'a, v', E')$ .*

*Proof.* From the previous lemma,  $E'$  (resp.  $E$ ) contains exactly the colors associated with  $\{x, a\}$  of all the occurrences of  $x$  in  $w$  since the one that initiates  $(a, xu'a, v', E')$  (resp.  $(a, xua, v, E)$ ). As  $E' \subset E$ , the occurrence of  $x$  in  $w$  that initiates  $(a, xu'a, v', E')$  is after the one that initiates  $(a, xua, v, E)$ . □

Now we give the definition of the partial commutation that we apply on colored words to simulate the semi-commutation.

**Definition 4.14.** *Let  $\theta$  be a semi-commutation over an alphabet  $\Sigma$ . The partial commutation  $\rho_\theta$  is defined by:*

$$\rho_\theta = \{((x, c), (y, c')) \in \Sigma_c^2 \mid (x, y) \in \theta \cap \theta^{-1} \text{ or } (x \neq y \text{ and } c(y) \neq c'(x))\}.$$

From this definition and remarks above straight follows the next property:

**Property 4.15.** *Let  $\theta$  be a semi-commutation over an alphabet  $\Sigma$ ,  $w$  be a word over  $\Sigma$  and  $(x, c), (y, c')$  two occurrences of letters in  $\tau_{(\Sigma, \theta)}(w)$ . Then, we have:*

$$(((x, c), (y, c')) \in \rho_\theta) \Rightarrow ((x, y) \in \theta \cup \theta^{-1}).$$

We now show two essential properties of the coloring. They assure that two letters that should not commute in a word  $u$  but have different colors in the coloring of  $u$  or two letters that should commute in  $u$  but have the same color in the coloring of  $u$  cannot be neighbours in any word of the closure of  $u$  under the semi-commutation.

**Lemma 4.16.** *Let  $\theta$  be a semi-commutation over an alphabet  $\Sigma$ . For each word  $u$  of  $\Sigma^*$  and for each  $(x, y)$  of  $\theta \cap \theta^{-1}$ , the following property holds:*

$$\begin{aligned} (\tau_{(\Sigma, \theta)}(u) = u_1(y, c)u_2(x, c')u_3 \text{ with } c(x) \neq c'(y)) \\ \Downarrow \\ (u_2 = v_1(y, c_1)v_3 \text{ with } c_1(x) = c'(y)). \end{aligned}$$

*Proof.* Let us suppose that  $\tau_{(\Sigma, \theta)}(u) = u_1(y, c)u_2(x, c')u_3$  with  $c(x) \neq c'(y)$ . Let us denote by  $q_1, q_2, q_3$  and  $q_4$  the states of  $\tau_{(\Sigma, \theta)}$  such that we have  $\delta(q_0, \varphi_\Sigma(u_1)) = (q_1, u_1)$ ,  $\delta(q_1, y) = (q_2, (y, c))$ ,  $\delta(q_2, \varphi_\Sigma(u_2)) = (q_3, u_2)$ , and  $\delta(q_3, x) = (q_4, (x, c'))$ . By definition, we have  $c(x) = \min(\bar{E}_{q_2, xy})$  and  $c'(y) = \min(\bar{E}_{q_4, xy})$ .

If  $\varphi_\Sigma(u_2)$  does not contain any  $y$ , we get from Lemma 4.10 the equality:  $E_{q_2, xy} = E_{q_3, xy} = E_{q_4, xy}$ . But  $c(x) = \min(\bar{E}_{q_2, xy})$  is different from  $c'(y) = \min(\bar{E}_{q_4, xy})$ , so there is at least one occurrence of  $y$  in  $\varphi_\Sigma(u_2)$ . Let us consider the last one:  $u_2 = v_1(y, c_1)v_3$  with  $|\varphi_\Sigma(v_3)|_y = 0$ , and  $\delta(q_2, \varphi_\Sigma(v_1)y) = (q'_2, v_1(y, c_1))$ . From Lemma 4.10, we get immediately  $c_1(x) = \min(\bar{E}_{q'_2, xy}) = \min(\bar{E}_{q_3, xy}) = c'(y)$  (see Fig. 1). □

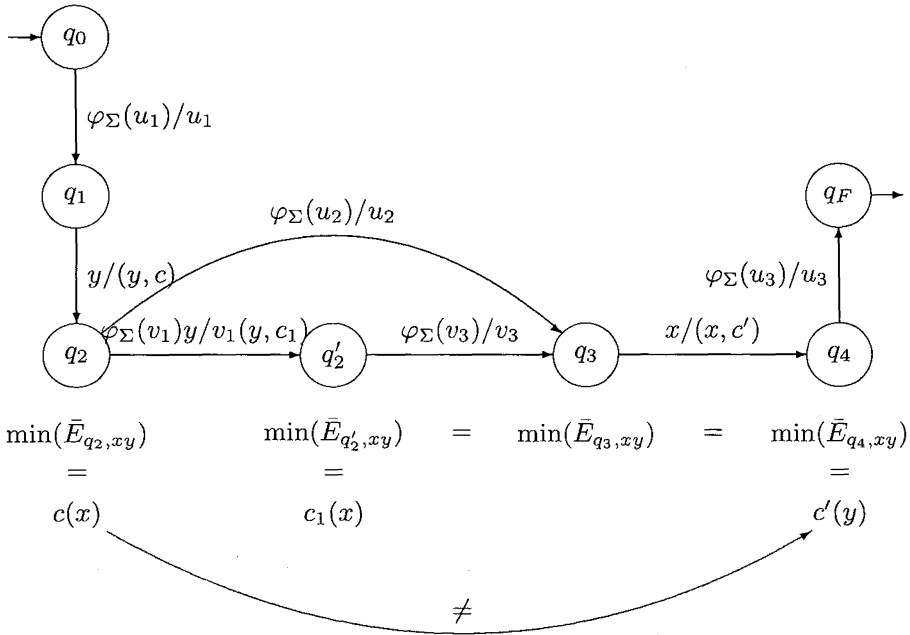


FIGURE 1. Proof of Lemma 4.16.

**Lemma 4.17.** *Let  $\theta$  be a semi-commutation over an alphabet  $\Sigma$ . For each word  $u$  of  $\Sigma^*$  and for each  $(x, y)$  of  $\theta \cap \bar{\theta}^{-1}$ , the following property holds:*

$$\begin{aligned}
 (\tau_{(\Sigma, \theta)}(u) = u_1(x, c)u_2(y, c')u_3 \text{ with } c(y) = c'(x)) \\
 \Downarrow \\
 (x\varphi_{\Sigma}(u_2)y \text{ is rigid for } xy).
 \end{aligned}$$

*Proof.* Let us denote by  $q_1, q_2, q_3$  and  $q_4$  the states of  $\tau_{(\Sigma, \theta)}$  such that we have  $\delta(q_0, \varphi_{\Sigma}(u_1)) = (q_1, u_1)$ ,  $\delta(q_1, x) = (q_2, (x, c))$ ,  $\delta(q_2, \varphi_{\Sigma}(u_2)) = (q_3, u_2)$ , and also  $\delta(q_3, y) = (q_4, (y, c'))$  (see Fig. 2).

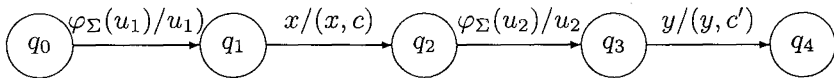


FIGURE 2. Proof of Lemma 4.17.

Let us suppose that  $xy$  belongs to  $\mathcal{T}_{(\Sigma, \theta)}$ , there exists a tuple  $(y, xy, E)$  in  $q_4$  where  $E$  contains exactly the set of colors associated with  $\{x, y\}$  of all the occurrences of  $x$  in  $u_1xu_2y$  (Lem. 4.12 I4.c), so  $E$  contains  $c(y)$ . As  $c'(x) = \min(\bar{E})$ , the equality  $c(y) = c'(x)$  is impossible, so  $xy$  does not belong to  $\mathcal{T}_{(\Sigma, \theta)}$ .

By definition of the transducer,  $c(y)$  is equal to  $\min(\bar{E}_{q_2,xy})$  and  $c'(x)$  is equal to  $\min(\bar{E}_{q_4,xy})$ . So  $q_4$  does not contain any tuple  $(y, xvy, w, E)$  such that  $c(y) = c'(x)$  belongs to  $E$ .

The  $x$  read between  $q_1$  and  $q_2$  initiates tuples  $(x, xvy, vy, \emptyset)$  for each word  $xvy$  of  $\mathcal{P}_{(\Sigma, \theta)}$ . Let us denote by  $x_1$  this occurrence of  $x$ . None of these tuples are represented in  $q_4$ : otherwise, their fourth components should contain  $c(y)$  (Lem. 4.12 II.c) and we just saw it is impossible. According to Definition 4.9, there are two possibilities for such tuples to be in a state and not to have any representation in the state reached after a transition.

In the first case, the representation of the tuple  $t$  we consider is  $(x, xvy, vy, \emptyset)$  in a state and the input letter is  $x$ . Then, this occurrence of  $x$  (denoted by  $x_2$ ) initiates a new tuple  $(x, xvy, vy, \emptyset)$  and  $t$  has no more representation. Then, according to Lemma 4.12 II.c, there is no occurrence of  $y$  between the occurrences  $x_1$  and  $x_2$  of  $x$ . We can simply consider the last occurrence of  $x$  in  $x\varphi_\Sigma(u_2)$  such that there is no  $y$  between this occurrence and the first one and consider the tuple  $(x, xvy, vy, \emptyset)$  initiated by this occurrence to come back to the second case.

In the second case, we have  $\varphi_\Sigma(u_2)y = v_1yv_3$  such that  $\delta(q_2, v_1) = (q, v'_1)$ ,  $\delta(q, y) = (q', c'')$ ,  $\delta(q', v_3) = (q_4, v'_3)$  and there exists in  $q$  a tuple  $(z, xvy, w, E)$  initiated in  $q_2$  that is no more in  $q'$ . There are two cases:

- $w = y$ , it means that  $v$  is a subword of  $v_1$ , so  $xvy$  is a subword of  $x\varphi_\Sigma(u_2)y$ ;
- there exists in  $q$  a tuple  $(z', xv'y, y, E')$  with  $E' \subset E$ . From Lemma 4.13, the occurrence of  $x$  that initiates this tuple is after the  $x$  that permits to reach  $q_2$ . So  $v'$  is a subword of  $v_1$  and  $xv'y$  is a subword of  $x\varphi_\Sigma(u_2)y$ .

□

Now, using the previous properties, we will show that the coloring we have defined satisfy our requirements, that is to say can be used to compute the closure under a semi-commutation.

**Theorem 4.18.** *Let  $\theta$  be a semi-commutation defined over an alphabet  $\Sigma$ . For each word  $u$  of  $\Sigma^*$ , we have:*

$$f_\theta(u) = \varphi_\Sigma(f_{\varrho_\theta}(\tau_{(\Sigma, \theta)}(u))).$$

*Proof.* Let  $v$  be a word of  $f_\theta(u)$ . Let us show by induction on the length of a minimal derivation from  $u$  to  $v$  that  $v$  belongs to  $\varphi_\Sigma(f_{\varrho_\theta}(\tau_{(\Sigma, \theta)}(u)))$ . This clearly holds when the length of the derivation is 0. Let us consider the derivation:

$$u \xrightarrow{\theta^n} w = w_1xyw_2 \xrightarrow{\theta} w_1yxw_2 = v.$$

By induction hypothesis, we have

$$\tau_{(\Sigma, \theta)}(u) \xrightarrow{\varrho_\theta^*} w'_1(x, c)(y, c')w'_2$$

with  $\varphi_\Sigma(w'_1)$  and  $\varphi_\Sigma(w'_2)$  respectively equal to  $w_1$  and  $w_2$ . If  $((x, c)(y, c'))$  does not belong to  $\varrho_\theta$  then, due to the definition of  $\varrho_\theta$ ,  $(x, y)$  does not belong to  $\theta^{-1}$



and  $c(y) = c'(x)$ . Since we consider a minimal derivation (according to Cor. 2.9), the two occurrences of letters that we consider are in the same order in  $\tau_{(\Sigma, \theta)}(u)$ , we have  $\tau_{(\Sigma, \theta)}(u) = u_1(x, c)u_2(y, c')u_3$ . According to Lemma 4.17,  $x\varphi_{\Sigma}(u_2)y$  is rigid for  $xy$  so these occurrences of  $x$  and  $y$  cannot be side by side in any word of  $f_{\theta}(u)$ . This lead to a contradiction. So,  $((x, c), (y, c'))$  belongs to  $\varrho_{\theta}$  and

$$\tau_{(\Sigma, \theta)}(u) \xrightarrow[\varrho_{\theta}]{*} w'_1(y, c')(x, c)w'_2 \in \varphi_{\Sigma}^{-1}(v).$$

We have the first part of the equality:  $f_{\theta}(u) \subseteq \varphi_{\Sigma}(f_{\varrho_{\theta}}(\tau_{(\Sigma, \theta)}(u)))$ .

Let  $v$  be a word of  $\varphi_{\Sigma}(f_{\varrho_{\theta}}(\tau_{(\Sigma, \theta)}(u)))$ . There exists a word  $v'$  belonging to  $f_{\varrho_{\theta}}(\tau_{(\Sigma, \theta)}(u))$  such that  $\varphi_{\Sigma}(v') = v$ . Let us show by induction on the length of a minimal derivation from  $\tau_{(\Sigma, \theta)}(u)$  to  $v'$  that  $v$  belongs to  $f_{\theta}(u)$ . This clearly holds when the length of the derivation is 0. Let us consider the derivation:

$$\tau_{(\Sigma, \theta)}(u) \xrightarrow[\varrho_{\theta}]{n} w' = w'_1(x, c)(y, c')w'_2 \xrightarrow[\varrho_{\theta}]{} w'_1(y, c')(x, c)w'_2 = v'.$$

By induction hypothesis, we have:

$$u \xrightarrow[\theta]{n} w = w_1xyw_2 \text{ with } \varphi_{\Sigma}(w'_1) = w_1 \text{ et } \varphi_{\Sigma}(w'_2) = w_2.$$

If  $(x, y)$  belongs to  $\theta$ , we have immediately

$$u \xrightarrow[\theta]{*} w = w_1xyw_2 \xrightarrow[\theta]{} v = \varphi_{\Sigma}(v').$$

We will show that the other case can never happen. Let us suppose that  $(x, y)$  do not belong to  $\theta$ . As we know that  $((x, c), (y, c'))$  belongs to  $\varrho_{\theta}$ , we deduce from Property 4.15 that  $(y, x)$  belongs to  $\theta$  and from the definition of  $\varrho_{\theta}$  that  $c(y) \neq c'(x)$ . Since the derivation is minimal, the occurrences  $(x, c)$  and  $(y, c')$  that we consider are in the same order in  $\tau_{(\Sigma, \theta)}(u)$ , we have  $\tau_{(\Sigma, \theta)}(u) = u_1(x, c)u_2(y, c')u_3$  so, according to Lemma 4.16,  $u_2 = v_1(x, c_1)v_2$  with  $c_1(y) = c'(x)$ . Thus, in each word of  $f_{\varrho_{\theta}}(\tau_{(\Sigma, \theta)}(u))$ , the two occurrences  $(x, c)$  and  $(y, c')$  that we consider will ever be separated by a factor containing  $(x, c_1)$  since neither  $((x, c), (x, c_1))$  belongs to  $\varrho_{\theta}$  nor  $((x, c_1), (y, c'))$ . This leads to a contradiction, the couple  $(x, y)$  must belong to  $\theta$ . So,  $v$  belongs to  $f_{\theta}(u)$  and as conclusion,  $\varphi_{\Sigma}(f_{\varrho_{\theta}}(\tau_{(\Sigma, \theta)}(u)))$  is included in  $f_{\theta}(u)$ . □

## 5. CASE OF REGULAR LANGUAGES

### 5.1. A SUFFICIENT CONDITION FOR THE TRANSDUCER TO BE FINITE

In this part, we give a sufficient condition for the image of a regular language by the transduction to be computed by a rational function. For this purpose we try to give a bound to values that can be contain by the sets “E” of the tuples of

the transducer. In the following of this section, we consider a semi-commutation  $\theta$  over a finite alphabet  $\Sigma$  and  $\tau_{(\Sigma, \theta)} = (\Sigma, \Sigma_c, Q, \delta, q_0, Q)$ . Let us first define the property we use:

**Definition 5.1.** *A language  $L$  over  $\Sigma$  satisfies the (P) property if and only if for each iterating factor  $u$  of  $L$ , each connected component in  $(\text{alph}(u), \bar{\theta})$  is strongly connected.*

**Lemma 5.2.** *Let  $L$  be a regular language over  $\Sigma$  verifying the (P) property. There exists an integer  $N_L > 0$  such that for each factor  $aub$  of  $L$  with  $(a, b)$  a couple of  $\theta \cap \bar{\theta}^{-1}$ :*

$$(\|aub\|_{ab} \geq N_L) \Rightarrow (aub \text{ is rigid for } ab).$$

*Proof.* Let  $n$  be the state number of the deterministic complete finite automaton of  $L$ . Let us set  $N_L = n \times \|\Sigma\| \times 2^{\|\Sigma\|}$ . If  $\|aub\|_{ab}$  is greater or equal to  $N_L$ , there exists a factorization:

$$aub = u_{1,1}u_{1,2} \dots u_{1,2^{\|\Sigma\|}} u_{2,1}u_{2,2} \dots u_{2,2^{\|\Sigma\|}} \dots u_{\|\Sigma\|,2^{\|\Sigma\|}}$$

with  $\|u_{i,j}\|_{ab}$  greater or equal to  $n$  for each  $i$  in  $\{1, 2, \dots, \|\Sigma\|\}$  and each  $j$  in  $\{1, 2, \dots, 2^{\|\Sigma\|}\}$ . So, at least one factor of each  $u_{i,j}$  is an iterating factor of  $L$  and contains at least one occurrence of  $a$  and one occurrence of  $b$ . From the (P) property of the language  $L$ , we deduce that each  $u_{i,j}$  has a strongly connected subword whose alphabet contains  $a$  and  $b$ . In other words, for each  $u_{i,j}$ , there exists a word  $au'_{i,j}b$  of  $\mathcal{P}_{\theta, \Sigma}$  such that  $\text{alph}(au'_{i,j}b)$  is included in  $\text{alph}(au_{i,j}b)$ . Since there exists at least  $\|\Sigma\|$  pairwise distinct couples  $(i, j)$  such that the alphabets of corresponding words  $u_{i,j}$  are identical,  $aub$  is rigid for  $ab$ .  $\square$

**Lemma 5.3.** *Let  $(a, b)$  belong to  $\theta \cap \bar{\theta}^{-1}$ ,  $u_1$  and  $u_2$  be words of  $\Sigma^*$ . We have:*

$$\begin{aligned} (\delta(q_0, u_1) = (q_1, u'_1) \text{ and } \delta(q_1, u_2) = (q_2, u'_2)) \\ \Downarrow \\ (\|E_{q_2, ab}\| \leq \|E_{q_1, ab}\| + \|u_2\|_{ab} + 1). \end{aligned}$$

*Proof.* Let us show the lemma by induction on the length of  $u_2$ . If  $u_2 = \varepsilon$  then  $\|E_{q_2, ab}\| = \|E_{q_1, ab}\|$ , so the relation is clearly satisfied.

Let us now suppose  $u_2 = v_2x$  with  $x$  a letter of  $\Sigma$ . Let us set  $(s_2, v'_2) = \delta(q_1, v_2)$ . We have  $\delta(s_2, x) = (q_2, x')$  and, by induction hypothesis, the following equation holds:

$$\|E_{s_2, ab}\| \leq \|E_{q_1, ab}\| + \|v_2\|_{ab} + 1.$$

If  $x$  is different from  $b$ , then  $\|u_2\|_{ab}$  is equal to  $\|v_2\|_{ab}$  and according to Lemma 4.10,  $E_{q_2, ab}$  is equal to  $E_{s_2, ab}$ , so the relation is satisfied.

If  $x$  is equal to  $b$  and  $\|u_2\|_{ab}$  to  $\|v_2\|_{ab} + 1$ , since  $\|E_{q_2, ab}\|$  is lower or equal to  $\|E_{s_2, ab}\| + 1$ , the relation is once again satisfied.

If  $x$  is equal to  $b$  and  $\|u_2\|_{ab}$  to  $\|v_2\|_{ab}$ , for each  $(y, a\alpha b, \beta, E)$  of  $s_2$ ,  $y$  is  $b$  and for each  $(y, ab, E)$  of  $s_2$ ,  $y$  is  $b$  thus we have:  $\|E_{q_2, ab}\| \leq \|E_{s_2, ab}\|$  and the relation is satisfied.  $\square$

**Definition 5.4.** Let  $q$  be a state of the transducer  $\tau_{(\Sigma, \theta)}$  and  $(a, b)$  a couple of  $\theta \cap \bar{\theta}^{-1}$ . The set  $\mathcal{S}_{q, ab}$  is defined by:

$$\mathcal{S}_{q, ab} = \{E \subseteq \mathbb{N} \mid \exists(x, aub, u', E) \in q\} \cup \{E \subseteq \mathbb{N} \mid \exists(x, ab, E) \in q\}.$$

**Lemma 5.5.** Let  $u$  be a word of  $\Sigma^*$  with  $\delta(q_0, u) = (q, u')$ . We have:

$$\forall(a, b) \in \theta \cap \bar{\theta}^{-1}, \forall E, F \in \mathcal{S}_{q, ab}, F \subseteq E \text{ or } E \subset F, \quad (1)$$

$$\|E_{q, ab}\| = \max_{F \in \mathcal{S}_{q, ab}} (\|F\|). \quad (2)$$

*Proof.* Let us first show the first property by induction of the length of  $u$ . If  $u$  is the empty word, then  $q = q_0 = \emptyset$  and the property is verified.

If  $u$  is equal to  $vx$  with  $x$  a letter of  $\Sigma$ , let us set  $(s, v') = \delta(q_0, v)$ . If  $x$  is different from  $b$  then each  $E$  belonging to  $\mathcal{S}_{q, ab}$  belongs to  $\mathcal{S}_{s, ab}$  or is empty. Thus, the property is satisfied. Now we have to consider the case when  $x$  is equal to  $b$ . Let us consider  $E$  and  $F$  belonging to  $\mathcal{S}_{q, ab}$ . Then, there exists  $E'$  and  $F'$  in  $\mathcal{S}_{s, ab}$  with  $E = E'$  and  $F = F'$  or  $E = E' \cup \min(\bar{E}_{s, ab})$  and  $F = F' \cup \min(\bar{E}_{s, ab})$ . Thus, the first property is always true.

The second property is a direct consequence of the first one.  $\square$

**Lemma 5.6.** Let  $(a, b)$  belong to  $\theta \cap \bar{\theta}^{-1}$  such that there exists a word  $a\alpha b$  in  $\mathcal{P}_{\theta, \Sigma}$ . Let  $u$  and  $v$  be words of  $\Sigma^*$  such that the word  $au$  does not contain any rigid left factor for  $ab$ . If  $\delta(q_0, vau) = (q, v'a'u')$  is a calculus of  $\tau_{(\Sigma, \theta)}$  then there exists a tuple  $(x, a\alpha b, \alpha', E)$  in  $q$  with  $\alpha'$  the shortest right factor of  $a\alpha b$  such that  $a\alpha b$  is a subword of  $au\alpha'$  and  $\|E\| = \|au\|_{ab}$ .

*Proof.* Let us show the lemma by induction on the length of  $u$ . If  $u$  is empty, then  $(a, a\alpha b, \alpha b, E = \emptyset)$  belongs to  $q$  and  $\alpha' = \alpha b$  is the shortest right factor of  $a\alpha b$  such that  $a\alpha b$  is a subword of  $au\alpha'$  and  $\|E\| = 0 = \|a\|_{ab}$ . Let us set  $u = wx$  with  $x$  a letter of  $\Sigma$  and  $\delta(q_0, vaw) = (s, v'a'w')$ . By induction hypothesis, there exists a tuple  $(y, a\alpha b, \alpha', E)$  in  $s$  satisfying the lemma.

If  $x$  is different from  $b$  then  $(z, a\alpha b, \alpha'/x, E)$  belongs to  $q$  and the lemma is satisfied.

If  $x$  is equal to  $b$ , let us suppose that there exists a tuple  $(z, a\beta b, b, F)$  in  $s$ . As  $awb$  has no left factor rigid for  $ab$ , we get  $E \subseteq F$ . Consequently, we have the same two cases, if there exists some  $(z, a\beta b, b, F)$  in  $s$  or not:

- when the last letter of  $\Pi_{ab}(aw)$  is  $b$  then  $(b, a\alpha b, \alpha', E)$  belongs to  $q$ ;
- otherwise,  $(b, a\alpha b, \alpha', E \cup \min(\bar{E}_{s, ab}))$  belongs to  $q$ .

In each case, the lemma holds.  $\square$

**Lemma 5.7.** *Let  $aub$  be a rigid word for  $ab$  that does not have any proper left factor rigid for  $ab$ . Then, considering the transducer  $\tau_{(\Sigma, \theta)}$ , we have:*

$$\forall v \in \Sigma^*, (\delta(q_0, vaub) = (q, w)) \Rightarrow (\|E_{q,ab}\| \leq \|aub\|_{ab}).$$

*Proof.* Let  $v$  be a word of  $\Sigma^*$  and  $\delta(q_0, vaub) = (q, w)$  be a calculus of  $\tau_{(\Sigma, \theta)}$ . Let us set  $\delta(q_0, v) = (q_1, v')$ ,  $\delta(q_1, a) = (q_2, a')$ , and  $\delta(q_2, u) = (q_3, u')$ . We have  $\delta(q_3, b) = (q, b')$ . Since  $aub$  is rigid for  $ab$ , there exists a subword  $\alpha$  of  $u$  such that  $a\alpha b$  belongs to  $\mathcal{P}_{\theta, \Sigma}$ .

The tuple  $(a, a\alpha b, \alpha b, \emptyset)$  belongs to  $q_2$  and, since  $aub$  does not contain any proper left factor rigid for  $ab$ , according to Lemma 5.6, there exists  $(x, a\alpha b, b, E)$  in  $q_3$  with  $\|E\| = \|au\|_{ab}$ . Moreover, according to Lemma 5.5, for each  $F$  of  $\mathcal{S}_{q_3, ab}$ , either  $F$  is a subset or is equal to  $E$ , either  $E$  is strictly included in  $F$ . According to the definition of  $\tau_{\Sigma, \theta}$ , there does not exist any tuple  $(x, a\alpha' b, \alpha'' b, F)$  such that  $E \subset F$  in  $q$ . Thus,  $\mathcal{S}_{q, ab}$  only contains sets  $F'$  such that:

- $\|F'\| \leq \|E\| + 1 = \|au\|_{ab} + 1 = \|aub\|_{ab}$  when the last letter of  $\Pi_{ab}(u)$  is  $a$ ;
- $\|F'\| \leq \|E\| = \|au\|_{ab} = \|aub\|_{ab}$  otherwise.

Since  $\|E_{q, ab}\|$  is equal to  $\max_{F' \in \mathcal{S}_{q, ab}} (\|F'\|)$ , according to Lemma 5.5, we have

$$\|E_{q, ab}\| \leq \|aub\|_{ab}.$$

□

**Lemma 5.8.** *Let  $L$  be a regular language over  $\Sigma$  satisfying the (P) property. There exists an integer  $K_L$  such that for each word  $u$  of  $L$  and for each couple  $(a, b)$  of  $\theta$ , considering the transducer  $\tau_{(\Sigma, \theta)}$ :*

$$(\delta(q_0, u) = (q, u')) \Rightarrow (\|E_{q, ab}\| \leq K_L).$$

*Proof.* Let  $N_L$  be the integer given by Lemma 5.2. We show that  $N_L + 1$  is suitable for  $K_L$  by induction on the length of the word  $u$ . Let  $(a, b)$  belong to  $\theta \cap \bar{\theta}^{-1}$ . If  $u$  is empty then  $E_{q_0, ab}$  is empty and the lemma holds.

Let us suppose that  $u = vx$  with  $x$  a letter of  $\Sigma$ . Let us set  $\delta(q_0, v) = (s, v')$ . We have  $\delta(s, x) = (q, x')$  and, by induction hypothesis,  $\|E_{s, ab}\| \leq K_L$ . If  $\|E_{s, ab}\| < K_L$ , we are done. Otherwise, we deduce from Lemma 5.3:  $\|v\|_{ab} \geq K_L - 1 = N_L$ . If  $x$  is different from  $b$ ,  $\|E_{q, ab}\|$  is equal to  $\|E_{s, ab}\|$  and we are done. Let us consider the case when  $u = vb$  and a factorization  $u_1 a u_2 b$  of  $u$  such that  $au_2 b$  is rigid for  $ab$  and has no proper left factor rigid for  $ab$ . Such a factorization exists according to Lemma 5.2 since  $\|v\|_{ab} \geq N_L$ . Moreover, since  $au_2 b$  has no proper left factor rigid for  $ab$ , we deduce from Lemma 5.2 that  $\|au_2 b\|_{ab}$  is lower or equal to  $N_L$ . Now, from Lemma 5.7, we have immediately:

$$\|E_{q, ab}\| \leq \|au_2 b\|_{ab} \leq N_L < K_L.$$

□

We are now able to enunciate the main result of this section.

**Proposition 5.9.** *Let  $\Sigma$  be an alphabet,  $\theta$  be a semi-commutation over  $\Sigma$  and  $L$  be a regular language over  $\Sigma$ . If  $L$  satisfies the (P) property then  $\tau_{(\Sigma,\theta)}(L)$  can be computed by a rational transduction.*

*Proof.* We have to show that a bounded number of states are reached in the computation of  $\tau_{(\Sigma,\theta)}(L)$ . For this purpose, it suffices to show that there exists an integer  $T_L$  such that for each  $(a, b)$  of  $\theta \cap \bar{\theta}^{-1}$  and for each word  $u$  of  $L$ :

$$(\delta(q_0, u) = (q, u')) \Rightarrow (\max(E_{q,ab}) \leq T_L).$$

Let us show that the integer  $K_L$  given by Lemma 5.8 is suitable for  $T_L$ .

Let us suppose that there exists a word  $u = vx$  with  $x$  a letter of  $\Sigma$  such that  $\delta(q_0, v) = (s, v')$ ,  $\delta(s, x) = (q, x')$ ,  $\max(E_{s,ab}) \leq K_L$ , and  $\max(E_{q,ab}) > K_L$ . From definition of  $\delta$ , we deduce that  $E_{s,ab} = \{0, 1, \dots, K_L\}$  but, in this case we have  $||E_{s,ab}|| = K_L + 1$ . This leads to a contradiction with Lemma 5.8.  $\square$

## 5.2. A NECESSARY CONDITION FOR THE TRANSDUCER TO BE FINITE

We now show that the condition of Proposition 5.9 is not only sufficient but also necessary for any rational coloring. In this part, we use a few technical intermediate lemmas that are proved using some formal languages notions which are quite different from the ones seen in the beginning of the paper. Thus, we need to introduce these notions.

**Notation 5.10.** *We denote by  $D_1'^*$  the semi-dyck language over one letter.*

**Notation 5.11.** *Let  $\theta$  be a semi-commutation over  $\Sigma$ . For any subalphabet  $X$  of  $\Sigma$ , we denote by  $(X, \bar{\theta})$  the restriction of the non commutation graph of this semi-commutation to  $X$ .*

**Definition 5.12.** *A language  $L$  is bounded if there exist words  $w_1, w_2, \dots, w_n$  such that  $L \subseteq w_1^* w_2^* \dots w_n^*$ .*

**Lemma 5.13.** *Let  $\theta$  be a semi-commutation over an alphabet  $\Sigma$  and  $L \subseteq \Sigma^*$  be a language which does not satisfy the (P) property. Then there exists a bounded language  $K$  included in  $L$  and a rational transduction  $\tau$  such that  $D_1'^* = \tau(f_\theta(K))$ .*

*Proof.* If the language  $L$  does not satisfy the (P) property then there exist three words  $x, y, u \in \Sigma^*$  and a subalphabet  $\alpha$  of  $\Sigma$  such that  $K = xu^*y \subseteq L$  and  $\alpha$  is a connected component of  $(\text{alph}(u), \bar{\theta})$  which is not strongly connected. Then, there exist two letters  $a$  and  $b$  in  $\alpha$  such that  $(b, a) \in \theta \cap \bar{\theta}^{-1}$  and there is no path from  $b$  to  $a$  in  $(\text{alph}(u), \bar{\theta})$ . Let us consider the two following alphabets:

- $\alpha_1 = \{b\} \cup \{z \in \text{alph}(u) \mid \text{there exists a path from } b \text{ to } z \text{ in } (\text{alph}(u), \bar{\theta})\}$ ;
- $\alpha_2 = \text{alph}(u) \setminus \alpha_1$ .

Note that for all  $(x, y)$  in  $\alpha_1 \times \alpha_2$ ,  $(x, y)$  belongs to  $\theta$  and  $\alpha_2$  is not empty since  $a \in \alpha_2$ .

Let us now introduce a new alphabet  $\bar{\Sigma} = \{\bar{a}, \bar{b}, \bar{x}, \bar{y}\}$  such that  $\bar{\Sigma} \cap \Sigma = \emptyset$  and two morphisms  $g$  and  $h$  defined from  $\bar{\Sigma}^*$  to  $\Sigma^*$  by:

- $g(\bar{a}) = a, g(\bar{b}) = b$  and  $g(\bar{x}) = g(\bar{y}) = \varepsilon$ ;
- $h(\bar{a}) = \Pi_{\alpha_1}(u), h(\bar{b}) = \Pi_{\alpha_2}(u), h(\bar{x}) = x$  and  $h(\bar{y}) = y$ .

Since the word  $\Pi_{\alpha_2}(u)\Pi_{\alpha_1}(u)$  belongs to  $f_\theta(u)$  and the word  $\Pi_{\alpha_2}(u)\Pi_{\alpha_1}(u)$  belongs to  $f_\theta(\Pi_{\alpha_1}(u)\Pi_{\alpha_2}(u))$  but  $\Pi_{\alpha_1}(u)\Pi_{\alpha_2}(u)$  do not belong to  $f_\theta(\Pi_{\alpha_2}(u)\Pi_{\alpha_1}(u))$ , we obtain

$$D_1'^* = g(h^{-1}(f_\theta(K)) \cap \bar{x}(\bar{a} + \bar{b})^*\bar{y}).$$

□

**Definition 5.14.** A family of languages is a rational cone if it is closed under rational transduction.

**Lemma 5.15.** Let  $\theta$  be a partial commutation over  $\Sigma$  and  $K$  be a bounded language. Then  $f_\theta(K)$  belongs to  $C_\cap(B)$ , the least rational cone closed under intersection containing all the bounded languages.

*Proof.* It is sufficient to prove the lemma when  $\theta$  is a partitioned commutation. A partial commutation  $\theta$  over  $\Sigma$  is a ( $k$ -)partitioned commutation if there exists a partition of the alphabet  $\Sigma : \{\Sigma_1, \dots, \Sigma_k\}$  such that

$$\theta = \bigcup_{\substack{i, j = 1 \\ i \neq j}}^k \Sigma_i \times \Sigma_j.$$

Indeed, if  $K$  is a bounded language and  $h$  a morphism,  $h(K)$  is a bounded language and it is shown in [4] the following result: for any partial commutation function  $f$ , there exist a morphism  $h$  and a partitioned commutation function  $f'$  such that  $f = h^{-1} \circ f' \circ h$ .

Let  $\{\Sigma_1, \dots, \Sigma_k\}$  be the corresponding partition of  $\Sigma$  and let us defined  $k$  copies of the alphabet  $\Sigma : X_1, \dots, X_k$ . Let  $h_i, i \in \{1, \dots, k\}$  be the corresponding bijections from  $\Sigma$  to  $X_i, i \in \{1, \dots, k\}$ . Then we clearly have

$$f_\theta(K) = \Pi_\Sigma \left( \bigsqcup_{i \in \{1, \dots, k\}} \{\Pi_{\Sigma_i}(u)h_i(u) \mid u \in K\} \bigcap \Sigma^* \left( \sum_{x \in \Sigma} h_1(x) \dots h_k(x) \right)^* \right).$$

Since  $K$  is a bounded language, the languages  $\{\Pi_{\Sigma_i}(u)h_i(u) \mid u \in K\}$  for  $i \in \{1, \dots, k\}$  are bounded languages too. Moreover, we know that  $C_\cap(B)$  is closed under shuffle since it is shown in [9] that every rational cone closed under intersection is closed under shuffle operation. It follows that  $f_\theta(K) \in C_\cap(B)$ .

□

We can now enunciate the main theorem of this section.

**Theorem 5.16.** *Let  $\theta$  be a semi-commutation over  $\Sigma$  and  $L \subseteq \Sigma^*$  be a regular language. There exist two rational functions  $h$  and  $g$  and a partial commutation  $\theta'$  over  $\Sigma'$  such that for each word  $u$  in  $L$ ,  $f_\theta(u) = g \circ f_{\theta'} \circ h(u)$  if and only if the language  $L$  satisfies the (P) property.*

*Proof.* From Proposition 5.9, we know that the condition is sufficient. Conversely, let us suppose that  $L$  does not satisfy (P) and there exist two rational functions  $h$  and  $g$  and a partial commutation  $\theta'$  over an alphabet  $\Sigma'$  such that for each word  $u$  in  $L$ ,  $f_\theta(u) = g(f_{\theta'}(h(u)))$ . We shall see that this leads to a contradiction. From Lemma 5.13, there exist a bounded language  $K \subseteq L$  and a rational transduction  $\tau$  such that  $D_1'^* = \tau(f_\theta(K))$ . Since  $h$  is a rational function, we get, from [8] and [10], that  $h(K)$  is a bounded language, then from Lemma 5.15, it follows that  $f_{\theta'}(h(K))$  belongs to  $\mathcal{C}_\cap(B)$  hence  $f_\theta(K) = g(f_{\theta'}(h(K)))$  belongs to  $\mathcal{C}_\cap(B)$ . It should follow that  $D_1'^* = \tau(f_\theta(K)) \in \mathcal{C}_\cap(B)$ , but this contradicts the following result enunciated in [12]:  $D_1'^*$  does not belong to the least rational cone containing the family of all the commutative languages which is equal to  $\mathcal{C}_\cap(B)$ .  $\square$

## 6. CONCLUSION

The results of this paper can be used to make a link between some results for semi-commutations and similar results for partial commutations. As conclusion, we present an example of this link.

Let us recall that there exists a sufficient condition to decide whether the closure of a regular language is regular. This condition is about the connexity of the iterating factors of the language and has been shown independently by Ochmański in [15] and Métivier in [14] for partial commutations and by Clerbout and Latteux in [5] for the semi-commutations.

**Theorem 6.1 (Métivier and Ochmański).** *Let  $\rho$  be a partial commutation over an alphabet  $\Sigma$  and  $L$  be a regular language over  $\Sigma$ . If each iterating factor of  $L$  is connected for  $\rho$  then the closure of  $L$  under  $\rho$  is regular.*

**Theorem 6.2 (Clerbout and Latteux).** *Let  $\theta$  be a semi-commutation over an alphabet  $\Sigma$  and  $L$  be a regular language over  $\Sigma$ . If each iterating factor of  $L$  is strongly connected for  $\theta$  then the closure of  $L$  under  $\theta$  is regular.*

Using Theorem 5.16, we propose a proof of Theorem 6.2 based on Theorem 6.1.

*Proof of Theorem 6.2.* Let  $\theta$  be a semi-commutation over an alphabet  $\Sigma$  and  $L$  be a regular language over  $\Sigma$  whose iterating factors are strongly connected for  $\theta$ . According to Proposition 5.9, the language  $\tau_{(\Sigma, \theta)}(L)$  is regular. So, it suffices to show that its iterating factors are connected for the partial commutation  $\rho_\theta$  to get the result.

Let  $u$  be an iterating factor of  $\tau_{(\Sigma, \theta)}(L)$ . Let us first remark that for each  $x$  and each  $y$  of  $\text{alph}(u)$  such that  $\varphi_\Sigma(x) = \varphi_\Sigma(y)$ ,  $(x, y)$  do not belong to  $\rho_\theta$ , so the sets of letters which have the same image under  $\varphi_\Sigma$  form cliques of the graph  $(\text{alph}(u), \bar{\rho}_\theta)$ . Since  $u$  is an iterating factor of  $\tau_{(\Sigma, \theta)}(L)$ ,  $\varphi_\Sigma(u)$  is an iterating factor

of  $L$ , so for each couple of letters  $x$  and  $y$  of  $\text{alph}(u)$ , there exists a path from  $\varphi_\Sigma(x)$  to  $\varphi_\Sigma(y)$  in the non-commutation graph  $(\text{alph}(\varphi_\Sigma(u)), \bar{\theta})$ . Let us denote by  $\varphi_\Sigma(x) = z_0, z_1, \dots, z_{n-1}, z_n = \varphi_\Sigma(y)$  the vertices of this path. Now, we just have to show that for each  $z_i$  and each  $z_{i+1}$  there exists in the graph  $(\text{alph}(u), \bar{\rho}_\theta)$  an edge between one  $z'_i$  and one  $z'_{i+1}$  such that  $\varphi_\Sigma(z'_i) = z_i$  and  $\varphi_\Sigma(z'_{i+1}) = z_{i+1}$ .

Since there exists an arc between  $z_i$  and  $z_{i+1}$ , two cases can happen. If  $(z_{i+1}, z_i)$  do not belong to the semi-commutation  $\theta$  then, by definition of the coloring, for each  $(z_i, c)$  and each  $(z_{i+1}, c')$  of the alphabet of  $u$ ,  $c(z_{i+1}) = 0$  and  $c'(z_i) = 0$  and we have an edge between  $(z_i, E)$  and  $(z_{i+1}, F)$ . If  $(z_{i+1}, z_i)$  belongs to  $\theta$  then it suffices to show that there exists a  $(z_i, c)$  and a  $(z_{i+1}, c')$  in the alphabet of  $u$  such that  $c(z_{i+1}) = c'(z_i)$ . This clearly holds: since  $u$  is a iterating factor of  $\tau_{(\Sigma, \theta)}(L)$ , we can consider  $u^2$ . In this word, for each letter  $(z_{i+1}, c')$  of the alphabet of  $u$ , at least one occurrence is preceded by a  $(z_i, c)$  (with no letter corresponding with a  $z_i$  between them). In this case, by definition of the coloring,  $c(z_{i+1}) = c'(z_i)$ , so there is an edge between these two letters. For each couple of letters  $x$  and  $y$  of  $\text{alph}(u)$ , there exists a path from  $x$  to  $y$  in  $(\text{alph}(u), \bar{\rho}_\theta)$ , therefore the word  $u$  is connected.

Since each iterating factor of  $\tau_{(\Sigma, \theta)}(L)$  is connected for  $\bar{\rho}_\theta$ , according Theorem 6.1,  $f_{\bar{\rho}_\theta}(\tau_{(\Sigma, \theta)}(L))$  is a regular language. According to Theorem 4.18,  $f_\theta(L)$  is the image by the strictly alphabetical morphism  $\varphi_\Sigma$  of  $f_{\bar{\rho}_\theta}(\tau_{(\Sigma, \theta)}(L))$ , so this language is regular.  $\square$

## REFERENCES

- [1] A. Arnold, An extension of the notions of traces and of asynchronous automata. *Theoret. Informatics Appl.* **25** (1991), 355–393.
- [2] M. Clerbout, *Commutations Partielles et Familles de Langages*. Ph.D. Thesis, Université des Sciences et Technologies de Lille (1984).
- [3] M. Clerbout and D. Gonzalez, Atomic semi-commutations. *Theoret. Comput. Sci.* **123** (1994) 259–272.
- [4] M. Clerbout and M. Latteux, Partial commutations and faithful rational transductions. *Theoret. Comput. Sci.* **34** (1984) 241–254.
- [5] M. Clerbout and M. Latteux, Semi-commutations. *Inform. and Comput.* **73** (1987) 59–74.
- [6] M. Clerbout, M. Latteux, Y. Roos and W. Zielonka, Semi-commutations and rational expressions, in *Proc. 19th International Colloquium on Automata Languages and Programming (ICALP'92)*, edited by W. Kuich. Springer, *Lecture Notes in Comput. Sci.* **623** (1992) 113–125.
- [7] V. Diekert and G. Rozenberg, *The Book of Traces*. World Scientific, Singapore (1995).
- [8] C.C. Elgot and J.E. Mezei, On relations defined by generalized finite automata. *IBM J. Res. Develop.* **9** (1965) 47–68.
- [9] S. Ginsburg and S.A. Greibach Principal AFL. *J. Comput. System Sci.* **4** (1970) 308–338.
- [10] S. Ginsburg, *The Mathematical Theory of Context-Free Languages*. McGraw-Hill (1966).
- [11] J.F. Husson, *Modélisation de la Causalité par des Relations d'Indépendance*. Ph.D. Thesis, Université Paul Sabatier, Toulouse (1996).
- [12] M. Latteux, Cônes rationnels commutativement clos. *Theoret. Informatics Appl.* **11** (1977) 29–51.
- [13] A. Mazurkiewicz, Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University (1977).



- [14] Y. Métivier, Une condition suffisante de reconnaissabilité dans un monoïde partiellement commutatif. *Theoret. Informatics Appl.* **20** (1986) 121–127.
- [15] E. Ochmański, Regular behaviour of concurrent systems. *Bull. Eur. Assoc. Theoret. Comput. Sci. EATCS* **27** (1985) 56–67.
- [16] E. Ochmański and P.A. Wacrenier, On regular compatibility of semi-commutations, in *Proc. 20th International Colloquium on Automata, Languages and Programming (ICALP'93)*, edited by A. Lingas, R. Karlsson and S. Carlsson. Springer, *Lecture Notes in Comput. Sci.* **700** (1993) 445–456.
- [17] W. Zielonka, Notes on finite asynchronous automata. *Theoret. Informatics Appl.* **21** (1987) 99–135.

Communicated by A. Arnold.

Received June 21, 2000. Accepted September 21, 2000.