

# Generalized derivatives

Lila KARI\*

*Academy of Finland and Department of Mathematics,  
University of Turku, 20500 Turku, Finland*

**Abstract.** The customary language-theoretic derivative of a word  $u$  with respect to a word  $v$  means the deletion of  $v$  from the beginning or end of  $u$ . We investigate the natural generalization, where  $v$  can be deleted from an arbitrary position in  $u$ . Apart from general closure and decidability properties, we pay special attention to regular languages, obtaining an exhaustive characterization.

## 1. Introduction

The left (right) quotient is a very basic language operation central to formal language theory. For example the derivatives, which are particular cases of quotients, are closely connected to the minimal finite deterministic automata (see [3]), the quotient has various interconnections with other operations, it can lead to interesting decision problems, and so on. Moreover, the left (right) quotient appears in some practical applications of formal language theory. For instance, under some language representations of images, zooming is accomplished with the quotient operation. See [2] for details.

A natural generalization of the left (right) quotient is the *deletion* operation defined in [6]. The deletion of a word  $v$  from a word  $u$  consists of erasing  $v$  not only from the left (right) extremity of  $u$  but from an arbitrary place in  $u$ . If  $v$  is not a subword of  $u$ , the result of the deletion is the empty set. The operation can be extended to languages in the obvious fashion. Deletion can be viewed as a one step rewriting relation of a special semi-Thue system (see [1], [5], [7] for details).

In this paper we will consider only the particular case in which the language to be deleted is a singleton. The resulting operation, called *derivative*, generalizes both left and right derivative operations.

Besides closure properties of the families of the Chomsky hierarchy under derivatives we investigate some properties of the derivatives of regular languages.

A sufficient condition under which a language gives rise to the same derivative with respect to two different words is obtained. Moreover, it is shown that the language consisting of all words, with respect to which a given regular language has the same derivative, is regular.

Also languages with maximal number of derivatives are studied. The main result shows that, for each integer  $n$ , there exists a minimal finite deterministic automaton with  $n$  states, which recognizes a language with maximal number of derivatives.

---

\*The work reported here is a part of the project 11281 of the Academy of Finland

Finally, the decision problem "Given languages  $L$  and  $R$ , does there exist a word  $w$  such that  $\partial_w L = R$ ?" is considered. The problem turns out to be decidable for regular languages  $L$  and  $R$ , but undecidable for more complex families of languages.

## 2. Basic definitions and notations

For a vocabulary  $\Sigma$ , we denote by  $\Sigma^*$  the free monoid generated by  $\Sigma$  under the concatenation operation; the null element of  $\Sigma^*$  is  $\lambda$  and  $\text{lg}(v)$  denotes the length of the word  $v \in \Sigma^*$ . For other notations and formal language theory notions like left (right) quotient, minimal finite deterministic automaton, generalized sequential machine, linear erasing morphism, the reader is referred to [9].

In studying the left and right quotient as operations on languages, of special interest is the case where the language to be deleted is a singleton.

The *left derivative* of a language  $L$  over  $\Sigma$  with respect to a word  $w \in \Sigma^*$  is obtained as a particular case of left quotient:

$$\partial_w^l L = \{u \in \Sigma^* \mid wu \in L\}.$$

The *right derivative* of the language  $L$  with respect to the word  $w$  is defined similarly as:

$$\partial_w^r L = \{u \in \Sigma^* \mid uw \in L\}.$$

A natural generalization of the right and left derivative is the operation where the word  $w$  is extracted not from the left or right extremity of a word in  $L$  but from an arbitrary place in it.

**Definition 1** *Let  $L$  be a language and  $w$  be a word over the alphabet  $\Sigma$ . The generalized derivative (shortly, derivative) of  $L$  with respect to  $w$  is defined as:*

$$\partial_w L = \{uv \in \Sigma^* \mid u w v \in L\}.$$

**Example 1** Let  $L = \{abbbab, aaabbb, abab\}$  and  $w = ab$ . The derivative of  $L$  with respect to  $w$  is:

$$\partial_w L = \{bbab, abbb, aabb, ab\}$$

whereas the left and right derivatives are respectively:

$$\partial_w^l L = \{bbab, ab\}, \quad \partial_w^r L = \{abbb, ab\}.$$

□

The derivative is a nondeterministic version of left (right) derivative in the sense that the left (right) derivative  $\partial_w^l u$  ( $\partial_w^r u$ ) consists of at most one word, whereas the derivative  $\partial_w u$  is in general a set of words of cardinality greater than one.

Note that in the previous example,  $\partial_w L$  properly contains the union  $\partial_w^l L \cup \partial_w^r L$ . In general, we have that

$$\partial_w^l L \cup \partial_w^r L \subseteq \partial_w L.$$

The left and right derivative can be obtained using the derivative and a marker which forces the position of the deletion. Indeed, for a language  $L$  and a word  $w$  over  $\Sigma$  we have

$$\partial_w^l L = \partial_{\$w}(\$L),$$

$$\partial_w^r L = \partial_{w\$}(L\$),$$

where  $\$$  is a new symbol which does not belong to  $\Sigma$ .

### 3. Closure properties

In this section we investigate the closure properties of the families in the Chomsky hierarchy under derivative. In order to prove the closure of REG and CF under derivative, we show first that the derivative of a language can be attained by a gsm with erasing.

**Theorem 1** *Let  $L$  be a language and  $w$  be a nonempty word over the same alphabet  $\Sigma$ . There exists a gsm  $g$  such that*

$$\partial_w L = g(L).$$

*Proof.* Let  $w$  be of the form  $w = a_1 a_2 \dots a_n$ ,  $n \geq 1$ ,  $a_i \in \Sigma$  for  $1 \leq i \leq n$ . Construct the gsm with erasing:

$$\begin{aligned} g &= (\Sigma, \Sigma, \{s_i \mid 0 \leq i \leq n\}, s_0, \{s_n\}, P), \\ P &= \{s_0 a \longrightarrow a s_0 \mid a \in \Sigma\} \cup \\ &\quad \{s_i a_{i+1} \longrightarrow s_{i+1} \mid 0 \leq i \leq n-1\} \cup \\ &\quad \{s_n a \longrightarrow a s_n \mid a \in \Sigma\}. \end{aligned}$$

The gsm  $g$  satisfies the requested equality.

Indeed, given a word  $v \in L$  as an input, the gsm  $g$  works as follows. If  $w$  is a subword of  $v$  then the rules of the type  $s_i a_{i+1} \longrightarrow s_{i+1}$  erase an occurrence of  $w$  from  $v$ , while the rules of the type  $s_0 a \longrightarrow a s_0$  and  $s_n a \longrightarrow a s_n$  leave the other letters of  $v$  unchanged. The output in this case is therefore  $\partial_w v$ . If the input word  $v$  does not contain  $w$  as a subword, the final state of  $g$  cannot be reached and therefore no output is produced.  $\square$

**Corollary 1** *The families of regular and context-free languages are closed under derivative.*

The closure of CS under derivative is obtained by modifying the proof of Theorem 1 such that the involved gsm is  $\lambda$ -free.

**Theorem 2** *The family of context-sensitive languages is closed under derivatives.*

*Proof.* Let  $L$  be a context-sensitive language over an alphabet  $\Sigma$  and  $w = a_1 \dots a_n$ ,  $n \geq 1$ ,  $a_i \in \Sigma$ ,  $1 \leq i \leq n$ , be a word over the same alphabet. If  $w \in L$  then

$$\partial_w L = [\partial_w(L - \{w\})] \cup \{\lambda\}.$$

If  $w = \lambda$  then  $\partial_\lambda L = L$ . Therefore the theorem will hold if we prove that  $\partial_w L$  is context-sensitive for  $w$  nonempty and not belonging to  $L$ .

We can modify the proof of Theorem 1 such that the constructed gsm is  $\lambda$ -free. Indeed, let  $\#$  be a new symbol which does not occur in  $\Sigma$  and consider the gsm:

$$\begin{aligned} g &= (\Sigma, \Sigma \cup \{\#\}, \{s_0, s_1, \dots, s_n\}, s_0, \{s_n\}, P), \\ P &= \{s_0 a \longrightarrow a s_0 \mid a \in \Sigma\} \cup \\ &\quad \{s_i a_{i+1} \longrightarrow \# a_{i+1} \mid 0 \leq i \leq n-1\} \cup \\ &\quad \{s_n a \longrightarrow a s_n \mid a \in \Sigma\}. \end{aligned}$$

It is easy to see that if  $h : (\Sigma \cup \{\#\})^* \longrightarrow \Sigma^*$  is the morphism defined by  $h(\#) = \lambda$ ,  $h(a) = a$ ,  $\forall a \in \Sigma$  then:

$$h(g(L)) = \partial_w L.$$

As  $\lg(w) = n$ , for every word  $\alpha \in g(L)$  the following inequality holds:

$$\lg(\alpha) \leq (n + 1)\lg(h(\alpha))$$

which proves that  $h$  is an  $(n + 1)$ -linear erasing with respect to  $g(L)$ .

As CS is closed under  $\lambda$ -free gsm mapping and under linear erasing, it follows that it is closed under derivatives, too.  $\square$

## 4. Derivatives of regular languages

In this section, some properties of the derivatives of regular languages are dealt with. Before that, some supplementary notions will be introduced. Let  $L$  be a regular language and  $A = (S, \Sigma, s_0, F, P)$  a finite deterministic automaton that accepts it. For every word  $w \in \Sigma^*$  define the function  $f_w^A : S \rightarrow S$  as follows:

$$f_w^A(s) = s' \text{ iff } sw \Longrightarrow^* s' \text{ in } A.$$

The function is total. If the automaton is clear from the context, we will denote the function simply by  $f_w$ .

Let  $L$  be a language over an alphabet  $\Sigma$ . The relations  $\mathcal{E}_L$  and  $\equiv_L$  over  $\Sigma^*$ , referred to as the *equivalence* and the *congruence* relation induced by  $L$  are defined as follows.  $u\mathcal{E}_L w$  iff, for all  $y \in \Sigma^*$ ,  $uy$  is in  $L$  exactly when  $wy$  is in  $L$ .  $u \equiv_L w$  iff, for all  $x, y \in \Sigma^*$ ,  $xuy \in L$  exactly when  $xwy \in L$ . Details about the equivalence and congruence relations induced by languages can be found for example in [8], pp.27-31, [4], pp.65-67.

It is known that  $u\mathcal{E}_L w$  iff the left derivatives of  $L$  with respect to  $u$  and  $w$  coincide, that is,  $\partial_u^l L = \partial_w^l L$ . As regards the congruence relation,  $u \equiv_L w$  iff  $f_u = f_w$  in a minimal finite deterministic automaton that accepts  $L$ . (Here minimal refers to the number of states.) Obviously, if  $u \equiv_L w$  then  $\partial_u L = \partial_w L$ . The reverse implication does not hold, as proved by the following example.

**Example 2** Let  $L = (ababa)^*$  and  $w_1 = babaa$ ,  $w_2 = baaba$ ,  $w_3 = abaab$ ,  $w_4 = aabab$ . Then we have:

$$\partial_{w_i} L = (ababa)^+, \text{ for } i = 1, 2, 3, 4,$$

but  $w_i \not\equiv_L w_j$  for  $i \neq j$ . For instance,  $aw_1 baba \in L$  but  $aw_i baba$  does not belong to  $L$  for  $i \neq 1$ .  $\square$

The derivatives define an equivalence relation  $\mathcal{D}_L$  over  $\Sigma^*$  by  $u\mathcal{D}_L v$  iff  $\partial_u L = \partial_v L$ .  $\mathcal{D}_L$  is an equivalence relation with a "coarser" class division than  $\equiv_L$ : each equivalence class of  $\mathcal{D}_L$  consists of one or more classes of  $\equiv_L$ .

In the following a sufficient condition under which a language gives rise to the same derivative with respect to two different words will be given.

**Theorem 3** Let  $L$  be a regular language accepted by the deterministic automaton  $A = (S, \Sigma, s_0, F, P)$  and  $u, v$  words over  $\Sigma^*$ . If  $f_u = f_v$  then  $\partial_u L = \partial_v L$ .

*Proof.* Let  $\alpha$  be a word in  $\partial_u L$ . There exist  $\alpha_1, \alpha_2 \in \Sigma^*$  and  $w \in L$  such that  $\alpha = \alpha_1 \alpha_2$  and  $w = \alpha_1 u \alpha_2$ . Consequently, the derivation:

$$s_0 \alpha_1 u \alpha_2 \Longrightarrow^* s_1 u \alpha_2 \Longrightarrow^* s_2 \alpha_2 \Longrightarrow^* s_f, \quad s_f \in F,$$

exists in the automaton  $A$ .

According to the definition of  $f_u : S \rightarrow S$ , we have  $s_2 = f_u(s_1)$ . As  $f_u = f_v$  it follows that  $s_2 = f_v(s_1)$  and consequently the derivation  $s_1 v \Rightarrow^* s_2$  exists in  $A$ . Therefore the following derivation can be constructed in  $A$ :

$$s_0 \alpha_1 v \alpha_2 \Rightarrow^* s_1 v \alpha_2 \Rightarrow^* s_2 \alpha_2 \Rightarrow^* s_f, \quad s_f \in F.$$

We have used the derivation  $s_0 \alpha_1 u \alpha_2 \Rightarrow^* s_f$  where the subderivation  $s_1 u \Rightarrow^* s_2$  has been replaced by  $s_1 v \Rightarrow^* s_2$ . This proves that the word  $\alpha_1 v \alpha_2$  belongs to  $L$  which implies that  $\alpha_1 \alpha_2$  belongs to  $\partial_v L$ .

The reverse inclusion can be proved similarly. □

The converse of the theorem does not hold, as shown by the following example.

**Example 3** Let  $L = \{ab, ca\}$  and let  $A = (S, \Sigma, s_0, F, P)$  be a finite deterministic automaton that accepts it, where:

$$\begin{aligned} S &= \{s_0, s_1, s_2, s_3, s_4, s'\}, \\ F &= \{s_2, s_4\}, \\ \Sigma &= \{a, b, c\}, \\ P &= \{s_0 a \rightarrow s_1, s_0 b \rightarrow s', s_0 c \rightarrow s_3\} \cup \\ &\quad \{s_1 a \rightarrow s', s_1 b \rightarrow s_2, s_1 c \rightarrow s'\} \cup \\ &\quad \{s_2 a \rightarrow s', s_2 b \rightarrow s', s_2 c \rightarrow s'\} \cup \\ &\quad \{s_3 a \rightarrow s_4, s_3 b \rightarrow s', s_3 c \rightarrow s'\} \cup \\ &\quad \{s_4 a \rightarrow s', s_4 b \rightarrow s', s_4 c \rightarrow s'\} \cup \\ &\quad \{s' a \rightarrow s', s' b \rightarrow s', s' c \rightarrow s'\}. \end{aligned}$$

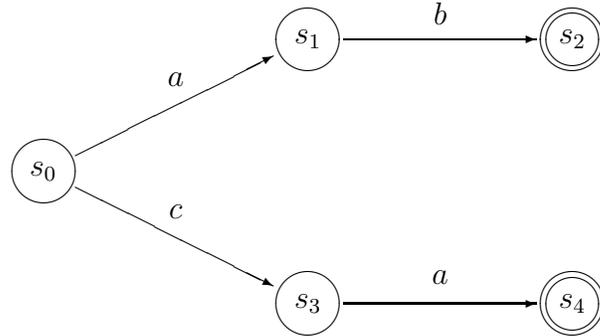


Figure 1: The automaton from Example 4.4.

The automaton  $A$  is represented in Figure 1. The state  $s'$  is a "garbage" state, introduced only to make the automaton deterministic. It has been omitted from the figure, for reasons of clarity.

The derivative of  $L$  with respect to both  $b$  and  $c$  equals  $\{a\}$  but the functions  $f_b$  and  $f_c$  are not equal:  $f_b(s_1) = s_2$  whereas  $f_c(s_1) = s'$ . □

**Corollary 2** *A regular language  $L$  has at most  $n^n$  different derivatives, where  $n$  is number of states in a minimal finite deterministic automaton accepting  $L$ .*

*Proof.* Let  $L$  be a regular language accepted by a minimal finite deterministic automaton  $A = (S, \Sigma, s_0, F, P)$  with  $n$  states. The number of different total functions  $f : S \rightarrow S$  is  $k = n^n$ . Using the previous theorem we deduce that there exist at most  $k$  different derivatives of  $L$ .  $\square$

**Example 4** Let us consider the minimal finite deterministic automaton  $A = (S, \Sigma, s_0, F, P)$  where

$$\begin{aligned} S &= \{s_0, s_1\}, \\ \Sigma &= \{a_1, a_2, a_3, a_4\}, \\ F &= \{s_0\}, \\ P &= \{s_0a_1 \rightarrow s_0, s_1a_1 \rightarrow s_1\} \cup \\ &\quad \{s_0a_2 \rightarrow s_0, s_1a_2 \rightarrow s_0\} \cup \\ &\quad \{s_0a_3 \rightarrow s_1, s_1a_3 \rightarrow s_1\} \cup \\ &\quad \{s_0a_4 \rightarrow s_1, s_1a_4 \rightarrow s_0\}. \end{aligned}$$

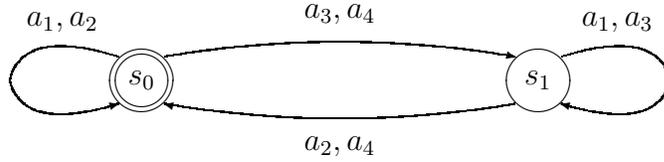


Figure 2: The automaton from Example 4.5.

The automaton  $A$  is represented in Figure 2.

The words  $a_1, a_2, a_3, a_4$  determine respectively the functions:

$$\begin{aligned} a_1 : f_1(s_0) &= s_0, & f_1(s_1) &= s_1, \\ a_2 : f_2(s_0) &= s_0, & f_2(s_1) &= s_0, \\ a_3 : f_3(s_0) &= s_1, & f_3(s_1) &= s_1, \\ a_4 : f_4(s_0) &= s_1, & f_4(s_1) &= s_0. \end{aligned}$$

According to the preceding corollary, the maximum number of different derivatives that  $L(A) = L$  can have is  $\text{card}(S)^{\text{card}(S)} = 4$ . In order to show that  $L$  has 4 different derivatives we will prove that  $\partial_{a_1}L, \partial_{a_2}L, \partial_{a_3}L, \partial_{a_4}L$  are all different.

The word  $a_3a_2a_1$  is in  $L$  therefore  $a_3a_1 \in \partial_{a_2}L$ . But  $a_3a_1$  is not in  $\partial_{a_1}L$  because neither  $a_1a_3a_1$  nor  $a_3a_1a_1$  belongs to  $L$ . Consequently,  $\partial_{a_2}L \neq \partial_{a_1}L$ .

The word  $a_1a_1$  belongs to  $L$  therefore  $a_1 \in \partial_{a_1}L$ . However,  $a_1$  is neither in  $\partial_{a_3}L$  nor in  $\partial_{a_4}L$  as none of the words  $a_1a_3, a_3a_1, a_1a_4, a_4a_1$  is in  $L$ . Consequently,  $\partial_{a_3}L \neq \partial_{a_1}L$  and  $\partial_{a_4}L \neq \partial_{a_1}L$ .

The word  $a_2a_1$  belongs to  $L$ , therefore  $a_1 \in \partial_{a_2}L$ . But, as none of the words  $a_3a_1, a_1a_3, a_4a_1, a_1a_4$  is in  $L$ ,  $a_1$  belongs neither to  $\partial_{a_3}L$  nor to  $\partial_{a_4}L$ . Consequently,  $\partial_{a_3}L \neq \partial_{a_2}L$  and  $\partial_{a_4}L \neq \partial_{a_2}L$ .

The word  $a_3a_4a_1$  belongs to  $L$ , therefore  $a_3a_1 \in \partial_{a_4}L$ . But  $a_3a_1$  does not belong to  $\partial_{a_3}L$  because neither  $a_3a_3a_1$  nor  $a_3a_1a_3$  is in  $L$ . Consequently,  $\partial_{a_3}L \neq \partial_{a_4}L$ .

The derivatives  $\partial_{a_1}L, \partial_{a_2}L, \partial_{a_3}L, \partial_{a_4}L$  are pairwise distinct. Consequently, the language  $L$  has four different derivatives which is the maximal number of derivatives it can have.  $\square$

Let  $A = (S, \Sigma, s_0, F, P)$  be a finite deterministic automaton. Two states  $s, s' \in S$  are called *distinguishable* if there exists a word  $u \in \Sigma^*$  such that  $su \xRightarrow{*} s_1, s'u \xRightarrow{*} s'_1$  and  $s_1 \in F, s'_1 \notin F$ , or viceversa. A finite deterministic automaton in which all states are distinguishable is minimal for its language (see [4], pp.68-71).

Using the method developed in the previous example we can prove a more general result.

**Theorem 4** *Let  $n$  be a natural number,  $n \geq 1$ . There exists a minimal finite deterministic automaton  $A_n$ , with  $n$  states, such that the language accepted by  $A_n$  has  $n^n$  different derivatives. Moreover, no other language accepted by a minimal finite deterministic automaton with  $n$  states has more different derivatives.*

*Proof.* Let  $n \geq 1$  be a natural number and  $A_n$  be the automaton:

$$\begin{aligned} A_n &= (S, \Sigma, s_1, F, P), \\ S &= \{s_1, s_2, \dots, s_n\}, \\ \Sigma &= \{f \mid f : S \rightarrow S\}, \\ F &= \{s_1\}, \\ P &= \{s_i f \rightarrow s_j \mid s_i, s_j \in S, f \in \Sigma, \text{ and } f(s_i) = s_j\}. \end{aligned}$$

Clearly,  $A_n$  is a finite deterministic automaton.  $A$  is also minimal. This follows because any two distinct states  $s_i$  and  $s_j$  are 1-distinguishable, i.e., a letter distinguishes them. Such a letter is  $f$  which, viewed as a function, maps  $s_i$  into  $s_1$  and  $s_j$  into  $s_2$ .

We shall show in the following that the language  $L = L(A_n)$  has  $n^n$  different derivatives.

If  $n = 1$  then  $\text{card}(\Sigma) = n^n = 1$ . The language accepted by the automaton  $A_1$  is  $L = \{f^p \mid p \geq 0\}$  and has one derivative,  $\partial_f L = L$ .

If  $n > 1$ , as  $\text{card}(\Sigma) = n^n$ , the proof is complete if we show that for every  $a, b \in \Sigma$ ,  $a \neq b$  we have  $\partial_a L \neq \partial_b L$ . Let  $a, b$  be two distinct letters in  $\Sigma$ . One of the following cases holds:

$$(i) \ a(s_1) \neq b(s_1).$$

If this is the case, then either  $a(s_1) \neq s_1$  or  $b(s_1) \neq s_1$ . Assume that the first alternative holds, the other one being similar. Choose  $f \in \Sigma$  with the following properties:

$$f(s_1) = s_1, f(a(s_1)) = a(s_1), f(b(s_1)) = s_1.$$

The word  $bf$  belongs to  $L$  as we can construct the derivation:

$$s_1 bf \xRightarrow{*} b(s_1) f \xRightarrow{*} f(b(s_1)) = s_1,$$

according to  $A_n$ . Consequently,  $f$  is a word in  $\partial_b L$ . However,  $f$  does not belong to  $\partial_a L$  as neither  $af$  nor  $fa$  belong to  $L$ :

$$\begin{aligned} s_1 af &\xRightarrow{*} a(s_1) f \xRightarrow{*} f(a(s_1)) = a(s_1) \neq s_1, \\ s_1 fa &\xRightarrow{*} f(s_1) a = s_1 a \xRightarrow{*} a(s_1) \neq s_1. \end{aligned}$$

Consequently, if (i) holds then  $\partial_a L \neq \partial_b L$ .

$$(ii) \ a(s_1) = b(s_1) \text{ and } a(s_i) \neq b(s_i) \text{ for some } 1 < i \leq n.$$

If this is the case then either  $a(s_i) \neq s_1$  or  $b(s_i) \neq s_1$ . Assume that  $a(s_i) \neq s_1$ , the other alternative being similar. We now consider two subcases.

$$(ii)' \ b(s_i) \neq s_i.$$

Choose  $f, g \in \Sigma$  with the properties:

$$\begin{aligned} f(x) &= s_i, \forall x \in S, \\ g(x) &= s_i, \forall x \in S, x \neq b(s_i) \text{ and } g(b(s_i)) = s_1. \end{aligned}$$

The word  $fbg$  belongs to  $L$  as we can construct the derivation:

$$s_1fbg \Longrightarrow f(s_1)bg = s_ibg \Longrightarrow b(s_i)g \Longrightarrow g(b(s_i)) = s_1,$$

according to  $A_n$ . This implies that  $fg$  belongs to  $\partial_b L$ . However,  $fg$  is not a word in  $\partial_a L$  as none of the words  $afg, fag, fga$  is in  $L$ :

$$s_1afg \Longrightarrow a(s_1)fg \Longrightarrow f(a(s_1))g = s_ig \Longrightarrow g(s_i) = s_i \neq s_1,$$

(we have used the fact that  $b(s_i) \neq s_i$ )

$$s_1fag \Longrightarrow f(s_1)ag = s_iag \Longrightarrow a(s_i)g \Longrightarrow g(a(s_i)) = s_i \neq s_1,$$

(we have used the fact that  $a(s_i) \neq b(s_i)$ )

$$s_1fga \Longrightarrow f(s_1)ga = s_iga \Longrightarrow g(s_i)a = s_ia \Longrightarrow a(s_i) \neq s_1,$$

(we have used the facts that  $b(s_i) \neq s_i$  and  $a(s_i) \neq s_1$ ).

Consequently, if  $(ii)'$  holds then  $\partial_a L \neq \partial_b L$ .

$(ii)''$   $b(s_i) = s_i$ .

As  $s_i \neq s_1$ , the above equality implies  $b(s_i) \neq s_1$ . As  $a(s_i) \neq b(s_i)$  and  $b(s_i) = s_i$  we deduce that  $a(s_i) \neq s_i$ . Therefore we are now in the case  $b(s_i) \neq s_1$  and  $a(s_i) \neq s_i$ , which is symmetric to  $(ii)'$  (with  $a$  and  $b$  switching their roles). Consequently, also if this case holds we obtain  $\partial_a L \neq \partial_b L$ .

As, in all the possible cases we found that  $\partial_a L \neq \partial_b L$ , we deduce that the two derivatives are distinct. The two letters  $a, b$  were arbitrarily chosen from  $\Sigma$ , therefore we conclude that all the  $n^n$  elements of  $\Sigma$  produce derivatives which are pairwise distinct. Consequently,  $L = L(A_n)$  has  $n^n$  different derivatives. The second claim of the theorem follows from Corollary 2.  $\square$

The following theorem shows that the language consisting of the words with respect to which a given regular language has the same derivative is regular.

**Theorem 5** *Let  $L$  be a regular language over the alphabet  $\Sigma$ . For any word  $w \in \Sigma^*$  the language:*

$$L_w = \{v \in \Sigma^* \mid \partial_w L = \partial_v L\}$$

*is regular and can be effectively constructed.*

*Proof.* Let  $A = (S, \Sigma, s_0, F, P)$  be a finite deterministic automaton that accepts the language  $L$ .

For every state  $s \in S$  and every function  $f : S \rightarrow S$  define:

$$L_{s,f} = \{w \in \Sigma^* \mid sw \Longrightarrow^* f(s)\}$$

and

$$L_f = \bigcap_{s \in S} L_{s,f}.$$

Each of the languages  $L_{s,f}$  is regular and each  $L_f$  is regular as a finite intersection of regular languages.

**Claim.**

$$L_f = \{w \in \Sigma^* \mid f_w = f\}.$$

”  $\subseteq$  ” Let  $w$  be a word in  $L_f$ . As  $w \in L_{s,f}$  for every state  $s \in S$ , the derivation  $sw \Longrightarrow^* f(s)$  exists in the automaton  $A$  for every  $s \in S$ .

According to the definition of  $f_w : S \rightarrow S$ , the derivation  $sw \Longrightarrow^* f_w(s)$  exists in  $A$  for every  $s \in S$ . As the automaton  $A$  is a deterministic one, we deduce that  $f(s) = f_w(s)$  for every state  $s \in S$ , that is,  $f = f_w$ .

”  $\supseteq$  ” Let  $w \in \Sigma^*$  be a word such that  $f(s) = f_w(s)$  for every  $s \in S$ . Then, for every state  $s \in S$  we have:

$$w \in L_{s,f} = \{w \in \Sigma^* \mid sw \Longrightarrow^* f(s) = f_w(s)\}$$

that is,

$$w \in \bigcap_{s \in S} L_{s,f} = L_f,$$

and the equality is proved.

The claim shows that the family  $\{L_f\}_{f:S \rightarrow S}$  determines a finite partition of  $\Sigma^*$  into disjoint regular languages  $L_f$ . To a set  $L_f$  belong those and only those words  $w$  such that  $f_w = f$ . To prove the theorem we show that for a given  $w \in \Sigma^*$ ,  $L_w$  is a union of some of the languages  $L_f$ .

There exist  $k = \text{card}(S)^{\text{card}(S)}$  different functions  $f : S \rightarrow S$ . Given a word  $w \in \Sigma^*$  we construct:

$$L' = \bigcup_{i=1}^k \{L_{f_i} \mid f_i : S \rightarrow S \text{ and } \exists v \in L_{f_i} : \partial_v L = \partial_w L\}.$$

The language  $L'$  is nonempty, containing at least the word  $w$ . We will prove in the following that  $L' = L_w$  where  $L_w$  is the language mentioned in the statement of the theorem.

Indeed, let  $u \in L'$ . There exist  $i \leq k$  and  $f_i : S \rightarrow S$  such that  $u \in L_{f_i}$  and  $\partial_v L = \partial_w L$  for some  $v \in L_{f_i}$ . According to the previous claim,  $f_u = f_v = f_i$ .

According to Theorem 3,  $\partial_u L = \partial_v L$  and therefore  $\partial_u L = \partial_v L = \partial_w L$ . This implies that  $u$  belongs to  $L_w$ .

For the reverse inclusion, let  $u$  be a word in  $L_w$ . There exists  $i \leq k$  such that the function  $f_u : S \rightarrow S$  equals the function  $f_i : S \rightarrow S$ . As, according to the definition of  $L_w$ ,  $\partial_u L = \partial_w L$  it follows that  $u$  belongs to the set

$$\{L_{f_i} \mid f_i : S \rightarrow S \text{ and } \exists u \in L_{f_i} : \partial_u L = \partial_w L\}$$

that is  $u$  belongs to  $L'$ . The equality  $L' = L_w$  is therefore proved. As  $L'$  is a regular language it follows that  $L_w$  is a regular language.

Using the above equality, for every word  $w$  the language  $L_w$  can be effectively constructed. Indeed, the sets  $L_{s,f}$ ,  $L_f$  can be constructed for every  $f : S \rightarrow S$  and every state  $s \in S$ .

In order to construct  $L'$  we use the remark that, for a total function  $f_i : S \rightarrow S$ , all the words in  $L_{f_i}$  give the same derivative with respect to  $L$ . This means that, for any function  $f_i : S \rightarrow S$  it suffices to check the equality  $\partial_v L = \partial_w L$  for an arbitrary word  $v \in L_{f_i}$ . If the answer is YES, the set  $L_{f_i}$  is taken into the union, else the function  $f_{i+1}$  is tried. The process terminates as the number of different functions to be checked is finite.

Note that REG is closed under derivative (see Corollary 1). The equivalence problem is decidable for regular languages that is, the problem ”Is  $\partial_v L$  equal with  $\partial_w L$ ?” is decidable for regular languages  $L$ . □

**Corollary 3** *Let  $L$  be a regular language over an alphabet  $\Sigma$ . For any word  $w \in \Sigma^*$  the languages:*

$$L_w^l = \{v \in \Sigma^* \mid \partial_w^l L = \partial_v^l L\},$$

$$L_w^r = \{v \in \Sigma^* \mid \partial_w^r L = \partial_v^r L\}$$

*are regular and can be effectively constructed.*

*Proof.* Let  $L$  be a language and  $w$  be a word over  $\Sigma$  and let  $\$$  be a symbol which does not occur in  $\Sigma$ . Consider  $L' = \$L$  (resp.  $L\$$ ) and  $w' = \$w$  (resp.  $w\$$ ). Then

$$L_w^l = (\$\setminus\{v \in (\Sigma \cup \{\$\})^* \mid \partial_{w'}^l L' = \partial_v^l L'\}) \cap \Sigma^*$$

$$\text{(resp. } L_w^r = (\{v \in (\Sigma \cup \{\$\})^* \mid \partial_{w'}^r L' = \partial_v^r L'\} / \$) \cap \Sigma^* \text{)}.$$

Using the preceding theorem and the fact that REG is closed under left (right) quotient and intersection with regular languages, we deduce that the languages  $L_w^l$  and  $L_w^r$  are regular and can be effectively constructed.  $\square$

## 5. Decision problems

In the previous section we have seen that there are finitely many languages that can be obtained from a regular one by derivative. Given two regular languages, a natural problem that arises is whether or not the second language can be obtained from the first by derivative.

In this section we investigate a more general problem, namely,

”Given languages  $L$  and  $R$ , does there exist a word  $w$  such that  $\partial_w L = R$ ?”.

The problem turns out to be decidable for regular languages  $L$  and  $R$  and undecidable for context-free languages  $L$  and regular languages  $R$  as well as for regular languages  $L$  and context-free languages  $R$ .

In order to show that the problem is decidable for regular languages  $L$  and  $R$ , an auxiliary result is needed.

**Theorem 6** *The (finitely many) languages that can be obtained from a regular one by derivative can be effectively constructed.*

*Proof.* Let  $L$  be a regular language accepted by a minimal finite deterministic automaton  $A = (S, \Sigma, s_0, F, P)$  with  $n$  states. According to Corollary 2, there exist at most  $k = n^n$  different derivatives of  $L$ .

For each  $i$ ,  $1 \leq i \leq k$ , consider the corresponding function  $f_i : S \rightarrow S$  and the language

$$L_{f_i} = \{w \in \Sigma^* \mid f_w = f_i\}.$$

As it has been seen in the proof of Theorem 5,  $L_{f_i}$  are regular languages and can be effectively constructed for all  $i$ ,  $1 \leq i \leq k$ .

We construct now the list containing all the derivatives of  $L$  as follows.

*Begin*

List( $q$ ) :=  $\emptyset$ ,  $1 \leq q \leq k$ ,

$i := 1, j := 1$

```

while  $i \leq k$  do
  if  $L_{f_i} \neq \emptyset$  then choose an arbitrary word  $w_i \in L_{f_i}$ ,
     $j := j + 1$ ,  $List(j) := \partial_{w_i} L$ 
  endif
endwhile
end

```

Note that, as the proof of Theorem 1 is constructive, we can effectively obtain the derivatives  $\partial_{w_i} L$  of the regular language  $L$ .

The list obtained in this way contains all the languages that can be possibly obtained from  $L$  by derivative. As the equivalence problem is decidable for regular languages, the duplicates can be eliminated from the list.  $\square$

**Corollary 4** *The problem "Does there exist a word  $w$  such that  $\partial_w L = R$ ?" is decidable for regular languages  $L$  and  $R$ .*

*Proof.* According to the preceding theorem, we can construct a list containing those and only those languages that can be obtained from  $L$  by derivative. If one of them equals  $R$ , the answer to our problem is YES, otherwise it is NO.

Note that in the case of an YES answer the proof of Theorem 6 effectively provides also the set of words

$$D = \{w \in \Sigma^* \mid \partial_w L = R\}.$$

Indeed, the set  $D$  can be obtained by taking the union of all those languages  $L_{f_i}$ ,  $1 \leq i \leq k$ , with the property  $\partial_{w_i} L = R$  for  $w_i \in L_{f_i}$ .  $\square$

Taking into account an observation made at the end of Section 2., we can rewrite the proofs of Theorem 6 and Corollary 4 for the left (right) derivatives obtaining the following known results.

**Theorem 7** *A regular language has finitely many left(right) derivatives. They can be effectively constructed.*

**Corollary 5** *The problem "Does there exist a word  $w$  such that  $\partial_w^l L = R$  (respectively  $\partial_w^r L = R$ )?" is decidable for regular languages  $L$  and  $R$ . One can also effectively construct the set*

$$D^l = \{w \mid \partial_w^l L = R\} \text{ (respectively } D^r = \{w \mid \partial_w^r L = R\}).$$

In the following, some undecidability results are proved. We begin with the simplest case, where the operation considered is the left (right) quotient.

**Theorem 8** *The problem "Does there exist a word  $w$  such that  $\partial_w^l L = R$ ?" is undecidable for context-free languages  $L$  and regular languages  $R$ .*

*Proof.* Let  $\Sigma$  be an alphabet,  $\text{card}(\Sigma) \geq 2$ , and let  $\#$  be a letter which does not occur in  $\Sigma$ . There exists a regular language  $R = \#\Sigma^*$  such that the problem of the theorem is undecidable for context-free languages  $L$ .

Indeed, the equation  $\partial_w^l(\#L) = \#\Sigma^*$  holds for languages  $L$  and words  $w$  over  $\Sigma$  exactly in the case  $w = \lambda$  and  $L = \Sigma^*$ . Hence, if we could decide the problem of the theorem, we would be deciding the problem "Is  $L = \Sigma^*$ ?" for context-free languages  $L$ , which is impossible.  $\square$

The problem "Does there exist a word  $w$  such that  $\partial_w^r L = R$ ?" is undecidable for context-free languages  $L$  and regular languages  $R$ . Indeed, if we take  $R = \Sigma^* \#$  and for a context-free  $L \subseteq \Sigma^*$  consider the language  $L\#$ , the proof is analogous to that of the preceding theorem.

**Theorem 9** *The problem "Does there exist a word  $w$  such that  $\partial_w L = R$ ?" is undecidable for context-free languages  $L$  and regular languages  $R$ .*

*Proof.* Let  $\Sigma$  be an alphabet,  $\text{card}(\Sigma) \geq 2$ , and let  $\#, \$$  be letters which do not occur in  $\Sigma$ . There exists a regular language  $R = \#\Sigma^*$  such that the problem of the theorem is undecidable for context-free languages  $L$ . We assume the contrary and show how to solve the problem "Is  $L = \Sigma^*$ ?" for context-free languages  $L$ .

Let  $L \subseteq \Sigma^*$  be a context-free language and consider the language  $\$\#L$ .

The equation

$$\partial_w(\#\Sigma^+\# \cup \$L\$) = \#\Sigma^+\# \cup \$\Sigma^*\$ \quad (*)$$

holds if and only if  $w = \lambda$  and  $L = \Sigma^*$ .

Consequently, if we could decide the problem of the theorem, we could decide whether for given context-free languages  $L$ , there exists a solution  $w$  to the equation (\*). This would, in turn, imply that we could decide the problem "Is  $L = \Sigma^*$ ?" for context-free languages  $L$ , which is impossible.  $\square$

## 6. Open problems

The generalization of the left (right) derivative defined in this paper is the simplest and most natural one. Some more sophisticated types of derivatives can be considered. For example, one can define the parallel derivative of  $u$  by  $w$  (all nonoverlapping occurrences of  $w$  are simultaneously deleted from  $u$ ), the permuted scattered derivative (the letters of  $w$  are deleted from  $u$  regardless of their positions) or the controlled derivative (the word  $w$  is deleted only if it appears after a "control letter" in  $u$ ). These and other generalizations of left (right) derivatives have been defined in [6],[10], and some of their properties have been investigated. For these operations, the study of problems analogous to the ones presented here for derivatives will be of interest.

## References

- [1] R.V.Book, M.Jantzen, C.Wrathall. Monadic Thue systems. *Theoretical Computer Science*, 19(1982), 231-251.
- [2] K.Culik, S.Dube. Rational and affine expressions for image description. *Discrete Applied Mathematics*, to appear. Tech.Report TR 9001, Univ. of South Carolina, 1990.
- [3] S. Eilenberg. *Automata, Languages and Machines*. Academic Press, London, 1974.
- [4] J.Hopcroft, J.Ulmann. *Introduction to Automata Theory, Languages and Computation*. Addison Wesley, Reading, Massachusetts, 1979.
- [5] M.Jantzen. Semi-Thue systems and generalized Church-Rosser properties. *Proc. Fete des Mots*, Rouen, France, (1982) 60-75.
- [6] L.Kari. *On Insertion and Deletion in Formal Languages*. Ph.D. Thesis, University of Turku, 1991.

- [7] M.Nivat, M.Benois. Congruences parfaites et quasi-parfaites. *Seminaire Dubreil 25<sup>e</sup> Annee* (1971-1972) 7-01-09.
- [8] A.Salomaa. *Theory of Automata*. Pergamon Press, Oxford, 1969.
- [9] A.Salomaa. *Formal languages*. Academic Press, London, 1973.
- [10] L.Santean. Six arithmetic-like operations on languages. *Revue Roumaine de Linguistique*, Tome XXXIII, 1988, Cahiers de linguistique theorique et applique, Tome XXV, No.1, Janvier-Juin, pp.65-73.