

# On language equations with invertible operations

Lila Kari

Academy of Finland and Department of Mathematics<sup>1</sup>  
University of Turku  
20500 Turku  
Finland

June 3, 2010

## Abstract

The paper studies language equations of the type  $X \diamond L = R$  and  $L \diamond Y = R$ , where  $L$  and  $R$  are given languages and  $\diamond$  is an invertible binary word(language) operation. For most of the considered insertion and deletion operations, the existence of both a solution and a singleton solution to these equations proves to be decidable for given regular  $L$  and  $R$ . In case  $L$  is a context-free language and  $R$  is a regular one, the existence of a solution is generally undecidable. The results can be extended to more complex linear equations, systems of linear equations as well as for equations of higher degree.

## 1 Introduction

Language and word equations have played a central role in formal language theory. Typical examples are the definition of a context-free language as the minimal solution of a system of equations (see [16]) or equations and systems of equations over free monoids or free semigroups (see [8] and its references),

In this paper we mainly study language equations of the form  $L \diamond Y = R$ ,  $X \diamond L = R$ , where  $\diamond$  is a binary word operation extended to languages in the obvious fashion.

The case where  $\diamond$  denotes catenation and the involved languages are regular has been considered by Conway in [3]. Some results relevant to our topic are presented in Section 2. If both languages  $L$  and  $R$  are regular, the existence of a solution to the equation is decidable and a maximal solution can be effectively constructed. (In case  $L$  is context-free, the existence of a solution is

---

<sup>1</sup>The work reported here is part of the project 11281 of the Academy of Finland

undecidable.) Moreover, for a given regular language  $R$ , one can provide a list of solutions to all possible equations  $LY = R$ , where  $L$  is an arbitrary language.

We are interested in solving language equations where the operations involved are more complex than catenation. Operations which generalize the catenation and quotient have been investigated in [9], [17], [11], [10]. Some of them are mentioned in Section 3: insertion, shuffle, controlled insertion, deletion, controlled deletion, scattered deletion, permuted deletion.

Solving equations of the type  $L \diamond Y = R$  for abstract language-theoretic operations  $\diamond$  which possess some kind of "inverses" bears some resemblance to questions studied for categories of abstract binary relations (see, for instance, [4]). One of the results states if  $\diamond$  is a binary operation possessing a "right-inverse" and a solution to the equation exists, then a maximal solution  $R'$  can be obtained.  $R'$  can be obtained from the given languages by applying the "right-inverse" of  $\diamond$ . A formal language theoretic formulation and proof of this result are given in Section 4 (Theorems 5 and 6).

After finding "right-inverses" and "left-inverses" of the operations defined in Section 3, the preceding results enable us to find solutions of equations involving insertion and deletion operations. The obtained results are natural extensions of the ones mentioned in Section 2 for catenation.

Sections 5, 6 deal with the decidability of the existence of solutions to the equations  $L \diamond Y = R$ , respectively  $X \diamond L = R$ .

For most operations, the problem turns out to be decidable for  $L$  and  $R$  regular languages. In these cases, the proofs are based on Theorems 5 and 6, and on the (effective) closure of the family of regular languages under the considered operations.

For many operations, the problem turns out to be undecidable for given context-free  $L$  and regular  $R$ . For these undecidability results no uniform approach is possible, as there is no powerful tool such as Theorem 5 to be applied in all the instances. Consequently, various ad-hoc methods, depending on the operation involved, are developed to reduce the problem to some known undecidable ones.

Section 7 points out how the results in Section 4 can be used to solve more general linear equations and systems of equations, as well as quadratic and other equations. In the end we summarize the problems which remain open and suggest some further directions of research.

## 2 The equations $LY = R$ and $XL = R$

Let  $\Sigma$  be a finite alphabet and  $\Sigma^*$  the set of all words over  $\Sigma$ , including the empty word  $\lambda$ . The length of a word  $w \in \Sigma^*$  is denoted by  $\text{lg}(w)$ . The left quotient of a word  $u$  by a word  $v$  is defined by

$$v \setminus u = w \text{ iff } u = vw,$$

and the right quotient of  $u$  by  $v$ ,

$$u/v = w \text{ iff } u = wv.$$

The mirror image of a word  $u$  is denoted by  $\text{Mi}(u)$ . For two languages  $L_1$  and  $L_2$  over  $\Sigma^*$ ,

$$L_1 - L_2 = \{u \mid u \in L_1 \text{ and } u \notin L_2\}, L_1^c = \Sigma^* - L_1.$$

REG denotes the family of regular languages. For unexplained formal language notions the reader is referred to [15].

In this section we investigate equations of the form  $LY = R$  and  $XL = R$ , where  $L, R$  are given languages,  $R$  regular.

**Theorem 1** *Let  $L, R$  be languages over the alphabet  $\Sigma$ ,  $R$  a regular one. If the equation  $LY = R$  has a solution  $Y \subseteq \Sigma^*$  then it has also a regular solution  $R'$ , which includes all the other solutions to the equation (set inclusion).*

*Proof.* Let  $R'$  be the language defined by:

$$R' = (L \setminus R^c)^c.$$

(i)  $R'$  is a regular language. Indeed, the left quotient of a regular language by an arbitrary language is regular.

(ii)  $LR' \subseteq R$ . Assume, for the sake of contradiction, that  $LR'$  is not included in  $R$ . There exist then words  $u \in L$ ,  $v \in R'$ , such that  $uv \in R^c$ . This implies that  $v = (u \setminus uv) \subseteq (L \setminus R^c)$  - a contradiction with the fact that  $v$  was a word in  $R'$ .

(iii) Any language  $Y$  with the property  $LY \subseteq R$  is included in  $R'$ . Indeed, assume that there exists a language  $Y$  as before such that  $Y - R' \neq \emptyset$ . Let  $v$  be a word in  $Y - R'$ . As  $v$  belongs to  $L \setminus R^c$ , there exist words  $w \in R^c$ ,  $u \in L$ , such that  $uw = v$ . This implies  $w \in LY \subseteq R$  - a contradiction with the fact that  $w$  was a word in  $R^c$ .

If the language equation  $LY = R$  has a solution  $Y \subseteq \Sigma^*$ , according to (iii),  $Y \subseteq R'$  and therefore  $R = LY \subseteq LR'$ . As, according to (ii), we have that  $LR' \subseteq R$ , we deduce that  $LR' = R$ . It has been showed in (i) that  $R'$  is a regular language, therefore the proof of the theorem is complete.  $\square$

**Corollary 1** *The regular solution  $R'$  from the preceding theorem can be effectively constructed if  $L$  is a regular or context-free language.*

In the following we will answer the question concerning whether or not the equation  $LY = R$  has a solution  $Y$ , where  $L, R$  are given languages,  $R$  a regular one. Moreover, the existence of a singleton solution, that is, a solution  $Y$  in the class of singleton languages, will be investigated.

More precisely, for given languages  $L$  and  $R$ ,  $R$  regular, we consider the problems:

"Does there exist a solution  $Y$  to the equation  $LY = R$ ?"

"Does there exist a singleton solution  $Y = \{w\}$  to the equation  $LY = R$ ?"

In the cases where the considered problem is decidable, it will follow from the proof that a solution to the equation can be effectively constructed.

**Theorem 2** *The problem "Does there exist a solution  $Y$  to the equation  $LY = R$ ?" is decidable for regular languages  $L$  and  $R$ .*

*Proof.* For given regular languages  $L, R$  over an alphabet  $\Sigma$  define:

$$R' = (L \setminus R^c)^c.$$

It has been proved in Theorem 1 that, if there exists a solution  $Y \subseteq \Sigma^*$  to the equation  $LY = R$ , then  $LR' = R$ . Moreover, the regular solution  $R'$  can be effectively constructed (see Corollary 1).

The algorithm which decides our problem will start with the construction of  $R'$ . Then we find out whether or not  $LR'$  equals  $R$ .  $\square$

**Example 1** Let  $L = \{a, ab\}$  and  $R = \{ab, abb\}$ . We are investigating the existence of a solution  $Y$  to the equation  $LY = R$ . Using the method of the preceding theorem we construct the languages:

$$\begin{aligned} R^c &= \{a, b\}^* - \{ab, abb\}, \\ L \setminus R^c &= \{a, b\}^* - \{b\}, \\ R' &= \{b\}. \end{aligned}$$

After checking the equality  $\{a, ab\}\{b\} = \{ab, abb\}$  we can answer positively to the question "Does there exist a language  $Y$  such that  $LY = R$ ?". Such a language is  $Y = R' = \{b\}$ .

In this particular situation  $R'$  is the only solution to our equation. This is not always the case. For example, if  $L = R = \Sigma^*$  then  $R^c = \emptyset$ ,  $L \setminus R^c = \emptyset$  and  $R' = \Sigma^*$ . However the language  $Y = \{\lambda\}$  also satisfies the equation  $\Sigma^*Y = \Sigma^*$ .

Note that if we take  $L = \{a, ab\}$  and  $R = \{ab, abb, ba\}$  we obtain the same  $R'$  as before, that is,  $R' = \{b\}$ . However, in this case the equality  $\{a, ab\}\{b\} = \{ab, abb, ba\}$  does not hold. According to the preceding theorem this implies that the equation  $\{a, ab\}Y = \{ab, abb, ba\}$  has no solutions.  $\square$

**Theorem 3** *The problem "Does there exist a singleton solution  $Y = \{w\}$  to the equation  $LY = R$ ?" is decidable for regular languages  $L$  and  $R$ .*

*Proof.* Let  $L, R$  be nonempty regular languages over an alphabet  $\Sigma$  and let  $m$  be the length of the shortest word in  $R$ . If there exists a word  $w$  such that  $L\{w\} = R$ , then it must satisfy the condition  $\lg(w) \leq m$ . The problem "Is  $L\{w\} = R$ ?" is decidable for words  $w$  and regular languages  $L$  and  $R$ .

The algorithm for deciding our problem will consist of checking whether or not  $L\{w\} = R$  for all words  $w$  with  $\text{lg}(w) \leq m$ . The answer is YES if such a word  $w$  is found, and NO otherwise.  $\square$

The study of the existence of a solution to the equation  $LY = R$ , when  $R$  is regular, is completed by the following undecidability results.

**Proposition 1** *The problem "Does there exist a solution  $Y$  to the equation  $LY = R$ ?" 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, we notice that the equation  $(L\#)Y = \Sigma^*\#$  holds for languages  $L, Y$  over  $\Sigma$  exactly in case  $L = \Sigma^*$  and  $Y = \{\lambda\}$ . 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.  $\square$

We notice that in the above proof the language  $Y = \{\lambda\}$  is a singleton. Therefore also the problem "Does there exist a singleton solution  $Y = \{w\}$  to the equation  $LY = R$ ?" is undecidable for context-free languages  $L$  and regular languages  $R$ .

We will conclude this section by showing that, for a given regular language  $R$ , one can effectively construct a list of solutions to all the possible equations  $LY = R$ , where  $L$  is an arbitrary language.

**Theorem 4** *Let  $R$  be a regular language over an alphabet  $\Sigma$ . There exists a finite number  $n \geq 1$  of distinct regular languages  $R'_i$ ,  $1 \leq i \leq n$ , such that for any  $L \subseteq \Sigma^*$  the following statements are equivalent:*

- (i) *There exists a solution  $Y \subseteq \Sigma^*$  to the equation  $LY = R$ .*
- (ii) *There exists an  $i$ ,  $1 \leq i \leq n$ , such that  $LR'_i = R$ .*

*Moreover, the regular languages  $R'_i$ ,  $1 \leq i \leq n$ , can be effectively constructed.*

*Proof.* It follows from Theorem 1. The languages  $R'_i$ ,  $1 \leq i \leq n$ , are constructed by forming the complements of all the possible (finitely many) languages that can be obtained from  $R^c$  by left quotient. Since the equivalence problem is decidable for regular languages, duplicates can be removed from the list  $R'_i$ .  $\square$

The list obtained in Theorem 4 may contain languages  $R'_i$  for which the equality  $LR'_i = R$  does not hold for any language  $L$ . However, these languages can be removed from the list as shown in the remaining part of this section.

Note that, by using the mirror image operator, results similar to Theorem 1, Corollary 1, Theorems 2, 3, Proposition 1, Theorem 4 can be obtained also

for equations of the type  $XL = R$ , where  $L, R$  are given languages and  $R$  is regular.

In particular, for a given regular language  $R$  one can effectively construct a finite list of distinct regular languages  $R''_1, R''_2, \dots, R''_m$ ,  $m \geq 1$  with the following property. For any language  $L$ , the equation  $XL = R$  has a solution  $X$  iff it has a solution among the languages  $R''_j$ ,  $1 \leq j \leq m$ .

We are now in position to effectively exclude from the list of Theorem 4 the languages  $R'_i$  for which the equality  $LR'_i = R$  does not hold for any  $L$ . According to the preceding property, if for a language  $R'_i$  such an  $L$  exists then we also have  $R''_j R'_i = R$  for some index  $j$ ,  $1 \leq j \leq m$ .

For each  $i$ ,  $1 \leq i \leq n$ , our algorithm will check, for all  $j$ ,  $1 \leq j \leq m$ , whether or not  $R''_j R'_i = R$ . If the equality holds for at least one index  $j$ , the language  $R'_i$  is retained in the list, otherwise it is eliminated.

In a similar way, we can effectively exclude from the list  $R''_1, \dots, R''_m$  the languages  $R''_j$  for which the equality  $R''_j L = R$  does not hold for any  $L$ .

### 3 Insertion and deletion operations

Catenation is a very basic binary word operation. As we will see in Sections 4 – 6 theorems similar to Theorems 1 – 4 and Proposition 1 hold also for more general binary word operations. Actually, as Theorem 5 will show, Theorem 1 can be generalized to concern any equation of the form  $L \diamond Y = R$  where the operation  $\diamond$  possesses an "inverse" operation.

In this section we will list some binary word operations for which theorems similar to Theorem 1 hold. For a more detailed study of these operations, see [9], [17]. The binary word operations are extended to languages in the natural fashion.

**Definition 1** *If  $\diamond$  is a binary word operation, we define the corresponding language operation by*

$$L_1 \diamond L_2 = \bigcup_{u \in L_1, v \in L_2} (u \diamond v).$$

The most natural generalization of catenation is the *insertion* operation. Given two words  $u$  and  $v$ , instead of catenating  $v$  at the right extremity of  $u$ , the new operation inserts it in an arbitrary place in  $u$ :

$$u \leftarrow v = \{u_1 v u_2 \mid u = u_1 u_2, u_1, u_2 \in \Sigma^*\}.$$

For example,  $cd \leftarrow a = \{acd, cad, cda\}$ , where  $a, c, d$  are letters in  $\Sigma$ . Notice that the result of insertion is a finite set of words and their catenation is an element of this set. Insertion can also be viewed as a one step rewriting relation of a semi-Thue system (see [7] for details).

A more exotic variant of insertion is obtained if we combine the insertion with the commutative variant. The commutative variant  $\text{com}(v)$  of a word  $v$  is the set of all words obtained by arbitrarily permuting the letters of  $v$ . The *permuted insertion* of  $v$  into  $u$  will then consist of inserting into  $u$  all the words from the commutative variant of  $v$ :

$$u \rightsquigarrow v = u \leftarrow \text{com}(v).$$

Observe that even though the above operations generalize the catenation, catenation cannot be obtained as a particular case of any of them. This happens because we cannot force the insertion to take place at the right extremity of the word. This brings up the notion of *control*: the insertion can be done only after a so-called control letter.

The *controlled insertion* of  $v$  into  $u$ , *next to the control-letter*  $a \in \Sigma$  (shortly, controlled insertion) is defined as

$$u \xleftarrow{a} v = \{u_1 a v u_2 \mid u = u_1 a u_2\}.$$

Special cases of catenation can be now obtained by using a marker and the controlled insertion next to the marker, and the general case by erasing the marker.

Notice finally that all the previously defined types of insertion were "compact". The word to be inserted was treated "as a whole". A "scattered" type of insertion can be considered as well. Instead of inserting a word, we sparsely insert its letters. If the inserted letters are in the same order as in the original, we obtain the well-known *shuffle* operation (see [14]):

$$u \amalg v = \{u_1 v_1 u_2 v_2 \dots u_k v_k \mid u = u_1 \dots u_k, v = v_1 \dots v_k, u_i, v_i \in \Sigma^*, k \geq 1\}.$$

Else, the *permuted scattered insertion* is obtained:

$$u \rightsquigarrow v = u \amalg \text{com}(v).$$

For each of the above mentioned variants of insertion, a "dual" deletion operation can be also considered. Take, for example, the *deletion* operation, which is the dual of the insertion operation. The deletion is the simplest and most natural generalization of left/right quotient. The deletion of  $v$  from  $u$  consists of erasing  $v$  not only from the left/right extremity of  $u$ , but from an arbitrary place in  $u$ :

$$u \rightarrow v = \{w \mid u = w_1 v w_2, w = w_1 w_2\}.$$

If  $v$  is not a subword of  $u$ , the result of the deletion is the empty set. Deletion can be viewed as a one step rewriting relation of a special semi-Thue system (see [1], [2] for details).

The following deletion operations are the counterparts of the insertion operations listed above. Properties of these operations and various related problems have recently been investigated in [11], [10], [12], [13].

The *permuted deletion* of  $v$  from  $u$  is

$$u \rightsquigarrow v = u \rightarrow \text{com}(v).$$

The *controlled deletion* of  $v$  from  $u$ , *next to the control-letter*  $a$ , (shortly, controlled deletion) is

$$u \xrightarrow{a} v = \{u_1 a u_2 \mid u = u_1 a v u_2\}.$$

The *scattered deletion* of  $v$  from  $u$  is

$$u \dashrightarrow v = \{u_1 u_2 \dots u_{k+1} \mid k \geq 1, u = u_1 v_1 u_2 v_2 \dots u_k v_k u_{k+1}, v = v_1 v_2 \dots v_k\}.$$

The *permuted scattered deletion* of  $v$  from  $u$  is  $u \rightsquigarrow v = u \dashrightarrow \text{com}(v)$ .

Finally, the *dipolar deletion* of the word  $v$  from the word  $u$  is the set consisting of the words obtained from  $u$  by erasing a prefix and a suffix whose catenation equals  $v$ :

$$u \rightleftharpoons v = \{w \in \Sigma^* \mid \exists v_1, v_2 \in \Sigma^* : u = v_1 w v_2, v = v_1 v_2\}.$$

## 4 Equations involving insertion and deletion operations

The process of solving the equation  $LY = R$  has much in common with the one of finding solutions to the algebraic equation  $a + y = b$ , where  $a, b$  are constants. In both cases, given the result of the operation and the left operand, the right operand could be recovered from them by using an "inverse" operation. In case of addition, this role is played by subtraction, and in case of catenation, the role is played by left quotient.

More precisely, the definition of left quotient states that for words  $u, v, w \in \Sigma^*$ , we have:

$$w = uv \text{ if and only if } v = u \setminus w.$$

In other words, given the result  $w$  of the catenation of  $u$  and  $v$ , and the left operand  $u$ , we can deterministically obtain the right operand by using the left quotient. Note that in the case of left quotient  $u \setminus w$ , we consider  $u$  to be the left operand.

As we are dealing also with operations whose result is a language instead of a word, the need arises for a more general definition of "inverse". Such an "inverse" operation will not solve the equation  $u \diamond Y = w$  but only loosely connect the right operand with the result and the left operand.



**Definition 2** Let  $\diamond, \square$  be two binary word operations. The operation  $\square$  is said to be right-inverse of the operation  $\diamond$  if for all words  $u, v, w$  over the alphabet  $\Sigma$  the following relation holds:

$$w \in (u \diamond v) \text{ iff } v \in (u \square w).$$

In other words, the operation  $\square$  is the right-inverse of the operation  $\diamond$  if, given a word  $w$  in the set  $u \diamond v$ , the right operand  $v$  belongs to a set obtainable from  $w$  and the other operand, by using  $\square$ . Notice that the relation "is the right-inverse of" is symmetric.

We are now ready to investigate the solutions to the equation  $L \diamond Y = R$ , in case  $\diamond$  possesses a right-inverse. The following result generalizes Theorem 1 by replacing catenation with abstract binary word (language) operation and left quotient with its right-inverse.

**Theorem 5** Let  $L, R$  be languages over an alphabet  $\Sigma$  and  $\diamond, \square$  be two binary word(language) operations right-inverses to each other. If the equation  $L \diamond Y = R$  has a solution  $Y$ , then also the language  $R' = (L \square R^c)^c$  is a solution of the equation. Moreover,  $R'$  includes all the other solutions of the equation (set inclusion).

*Proof.* We shall begin by proving two properties of the language  $R'$ .

(i)  $L \diamond R' \subseteq R$ . Assume the contrary and let  $w$  be a word belonging to  $L \diamond R'$  but not to  $R$ . There exist words  $u \in L, v \in R'$  such that  $w \in (u \diamond v)$ . As  $\square$  is the right-inverse of  $\diamond$ , we further deduce that  $v$  belongs to  $(u \square w)$  which is a subset of  $L \square R^c$ . We arrived at a contradiction, as  $v$  was a word in  $R'$ . Our assumption was false, therefore  $L \diamond R' \subseteq R$ .

(ii) Any language  $Y$  with the property  $L \diamond Y \subseteq R$  is included in  $R'$ . Assume the contrary and let  $v$  be a word belonging to such an  $Y$ , but not to  $R'$ . As the word  $v$  belongs to  $R'^c = L \square R^c$ , there exist words  $w \in R^c, u \in L$  such that  $v \in (u \square w)$ . As  $\square$  is the right-inverse of  $\diamond$ , we deduce that  $w$  is a word in  $u \diamond v$ . This implies that  $w$  belongs to  $L \diamond Y$  which was, according to the hypothesis, a subset of  $R$ . We arrived at a contradiction, as  $w$  was a word in  $R^c$ . Consequently, our assumption that such a language  $Y$  exists was false.

Return to the proof of the theorem. If there exists a solution  $Y$  to the language equation  $L \diamond Y = R$  then, according to (ii),  $Y \subseteq R'$ , which implies  $R = L \diamond Y \subseteq L \diamond R'$ . As (i) states that  $L \diamond R' \subseteq R$ , we conclude that  $L \diamond R' = R$ , that is,  $R'$  is also a solution of the equation.  $\square$

Observe that Theorem 1 can now be obtained as a consequence of the preceding theorem by using the closure properties of REG under catenation and quotient and the fact that left quotient is the right-inverse of catenation.

Theorem 5 gives a powerful tool for solving the equation  $L \diamond Y = R$ , when  $L$  and  $R$  are regular languages and  $\diamond$  possesses a right inverse. The following observation allows us to formulate Theorem 5 for equations as above, involving

operations defined in Section 3. Before that, we introduce the notion of reversing an operation.

**Definition 3** *Let  $\diamond$  be a binary word operation. The word operation  $\diamond^r$  defined by  $u \diamond^r v = v \diamond u$  is called reversed  $\diamond$ .*

**Observation:** The following operations are right-inverses to each other:

catenation	—	left quotient
insertion	—	reversed dipolar deletion
shuffle	—	reversed scattered deletion
right quotient	—	reversed left quotient
deletion	—	dipolar deletion
scattered deletion	—	scattered deletion

Also the operations of controlled insertion, controlled deletion, permuted insertion, permuted scattered insertion, permuted deletion, permuted scattered deletion possess right-inverses.  $\square$

As we have seen in Section 2, the results concerning the equation  $LY = R$  could be transferred without much difficulty to the equation  $XL = R$ . Analogously, the result of Theorem 5 can be modified to refer to equations  $X \diamond L = R$ . With this in mind, a notion corresponding to that of right-inverse has to be defined.

**Definition 4** *Let  $\diamond, \square$  be two binary word operations. The operation  $\square$  is said to be the left-inverse of the operation  $\diamond$  if, for all words  $u, v, w$  over the alphabet  $\Sigma$ , the following relation holds:*

$$w \in (u \diamond v) \text{ iff } u \in (w \square v).$$

In other words, the operation  $\square$  is the left-inverse of the operation  $\diamond$  if, given a word in  $u \diamond v$ , the left operand  $u$  belongs to the set obtained from  $w$  and the other operand  $v$  by using the operation  $\square$ . The relation "is the left-inverse of" is symmetric.

Note that the operation  $\square$  is the left-inverse of the operation  $\diamond$  if and only if the operation  $\square^r$  is the right-inverse of the operation  $\diamond^r$ .

Using the notion of the left-inverse, we are now ready to state a twin theorem of Theorem 5.

**Theorem 6** *Let  $L, R$  be languages over an alphabet  $\Sigma$  and  $\diamond, \square$  be two binary word(language) operations, left-inverses to each other. If the equation  $X \diamond L = R$  has a solution  $X \subseteq \Sigma^*$ , then also the language  $R' = (R^c \square L)^c$  is a solution of the equation. Moreover,  $R'$  includes all the other solutions of the equation (set inclusion).*

*Proof.* It follows from Theorem 5 by replacing  $\diamond$  with  $\diamond^r$  and  $\square$  with  $\square^r$ .  $\square$

The results concerning the solutions of the equation  $XL = R$ ,  $L, R$  regular, can be now obtained as consequences of the preceding theorem, as REG is closed under catenation and right quotient and the right quotient is the left-inverse of catenation.

The following observation allows us to investigate the solutions of the equation  $X \diamond L = R$  for the operations defined in Section 3

**Observation:** The following operations are left-inverses to each other:

catenation	—	right quotient
insertion	—	deletion
controlled insertion	—	controlled deletion
shuffle	—	scattered deletion
permuted insertion	—	permuted deletion
permuted scattered insertion	—	permuted scattered deletion
left quotient	—	reversed catenation
dipolar deletion	—	reversed insertion.

## 5 Solutions to the equation $L \diamond Y = L$

This section deals with the decidability of the question whether or not the equation  $L \diamond Y = R$  has a solution  $Y$ , where  $L, R$  are given languages,  $R$  a regular one, and  $\diamond$  is an binary insertion or deletion operation. Moreover, the existence of a singleton solution, that is, a solution  $Y$  in the class of singleton languages, will be investigated.

More precisely, for a binary language operation  $\diamond$  and for given languages  $L$  and  $R$ ,  $R$  regular, we consider the problems:

QY: "Does there exist a solution  $Y$  to the equation  $L \diamond Y = R$ ?"

QY': "Does there exist a singleton solution  $Y = \{w\}$  to the equation  $L \diamond Y = R$ ?"

*Note:* In case  $\diamond$  denotes a controlled operation the above problems have to be modified. For example, in the case of controlled insertion, QY becomes: "Do there exist a language  $Y$  and a control letter  $a$  such that  $L \xleftarrow{a} Y = R$ ?"

The problems turn out to be decidable in case all operands involved are regular and REG is closed under the operation  $\diamond$ . Moreover, in case a solution to the equation exists, it can be effectively constructed.

**Theorem 7** *Let  $\diamond$  be one of the operations: catenation, insertion, shuffle, controlled insertion, left/right quotient, deletion, scattered deletion, controlled deletion, dipolar deletion. Then the QY is decidable for regular languages  $L$  and  $R$ .*

*Proof.* Analogous to that of Theorem 2 and using the results from Theorem 5 and the proofs of the closure properties of REG under the above operations (see [9]) which are all constructive.  $\square$

Let  $\diamond$  denote one of the operations: catenation, shuffle, permuted insertion, permuted scattered insertion, controlled insertion. The proof of the Theorem 3 can be used to show that in all mentioned cases the problem  $QY'$  is decidable for regular languages  $L$  and  $R$ .

Let  $\diamond$  denote one of the operations: insertion, iterated insertion, shuffle, permuted scattered insertion, permuted insertion and controlled insertion. A proof similar to that of Proposition 1 can be used to show that in all the cases, the problems  $QY$  and  $QY'$  are undecidable for context-free languages  $L$  and regular languages  $R$ . (If  $\diamond$  stands for controlled insertion, we choose the control letter to be  $\#$ .)

If  $\diamond$  denotes a binary deletion operation and  $L$  is a given language, a word  $y$  is called *right-useful with respect to  $L$  and  $\diamond$*  if there exists an  $x \in L$  such that  $x \diamond y \neq \emptyset$ . A language  $Y$  is called right-useful with respect to  $L$  and  $\diamond$  if it consists only of right-useful words with respect to  $L$  and  $\diamond$ .

If  $L$  and  $\diamond$  are clear from the context, the word  $y$ , the language  $Y$  respectively the function  $\Delta$  will be termed simply *right useful*.

From the above definitions it follows that the problems  $QY$ ,  $QY'$  for deletion, are equivalent with the corresponding problems where the existence of a right useful language or word, is investigated. Therefore in the remaining part of this section, when we want to prove an undecidability result about deletion, we will mean a right useful language or word when referring to the corresponding items whose existence is studied. We begin with the simplest case, where the operation considered is the left (right) quotient.

**Proposition 2** *If  $\diamond$  denotes the left quotient, the problem  $QY$  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  $Y \setminus (\#L) = \#\Sigma^*$  holds for languages  $L$  and  $Y$  over  $\Sigma$  exactly in the case  $Y = \{\lambda\}$  and  $L = \Sigma^*$ . (Recall our convention concerning right usefulness.) 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.  $\square$

Notice that in the above proof the language  $Y = \{\lambda\}$  is a singleton. Therefore, if  $\diamond$  denotes the left quotient, the problem  $QY'$  is undecidable for context-free languages  $L$  and regular languages  $R$ .

Also in the case of the right quotient the problems  $QY$  and  $QY'$  are undecidable for context-free languages  $L$  and regular languages  $R$ . Indeed, if we take  $R = \Sigma^*\#$  and for a context-free  $L \subseteq \Sigma^*$ ,  $L = L\#$ , the proof is analogous to that of the preceding theorem.

**Proposition 3** *If  $\diamond$  denotes the deletion, the problem  $QY$  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  $R = \#\Sigma^+\# \cup \$\Sigma^*\$$  such that QY 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  $L = \#\Sigma^+\# \cup \$L'\$$ . For all languages  $Y \subseteq \Sigma^*$ , the equation:

$$\#\Sigma^+\# \cup \$L'\$ \rightarrow Y = \#\Sigma^+\# \cup \$\Sigma^*\$ \quad (*)$$

holds if and only if  $Y = \{\lambda\}$  and  $L' = \Sigma^*$ .

The implication " $\Leftarrow$ " is obvious. For the reverse implication assume that (\*) holds and that  $Y$  contains a nonempty word  $w$ . As we are considering only right useful words, one of the next situations must hold:  $w = \#$ ,  $w \in \Sigma^+$ ,  $w = \#v$ ,  $w = v\#$ ,  $w = \#v\#$ ,  $w = u\$$ ,  $w = \$u$ ,  $w = \$u\$$ ,  $v \in \Sigma^+$ ,  $u \in \Sigma^*$ . Each of these situations leads to a contradiction. For example  $w = \#v\#$ ,  $v \in \Sigma^+$  implies  $\#v\# \in L$ , and therefore,

$$\lambda \in (\#v\# \rightarrow \#v\#) \subseteq R,$$

which contradicts the form of the words in  $R$ . Consequently, our assumption that  $Y$  contains nonempty words was false.

The fact that  $Y = \{\lambda\}$  implies that  $L' = \Sigma^*$ , and the proof of the reverse implication is complete.

If we could decide the problem of the theorem, we could decide whether for given context-free languages  $L'$ , there exists a solution  $Y$  to the equation (\*). According to the facts proved above, this would in turn imply that we could decide the problem "Is  $L' = \Sigma^*$ ?" for context-free languages  $L'$ , which is impossible.  $\square$

Noticing that in the above proof  $Y = \{\lambda\}$  is a singleton language, the proof can be used to show that for  $\diamond$  denoting the deletion, the problem QY' is undecidable for context-free languages  $L$  and regular languages  $R$ .

A similar proof using the same construction can be used to show that for  $\diamond$  denoting the permuted deletion, the scattered deletion, the iterated deletion, the permuted scattered deletion the problems QY and QY' are undecidable for context-free languages  $L$  and regular languages  $R$ .

**Proposition 4** *If  $\diamond$  denotes the controlled deletion, the problem QY 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 QY 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  $L'$ .

For a given context-free language  $L' \subseteq \Sigma^*$ , construct  $L = L'\#\$$ .

Then we have:

$$L' \# \$ \xrightarrow{\#} Y = \Sigma^* \# \quad \text{iff} \quad Y = \{\$\}, \text{ and } L' = \Sigma^*. \quad (*)$$

The implication " $\Leftarrow$ " is obvious. For the reverse implication assume that  $Y$  contains at least a word  $w \neq \$$ . The only possibility is  $w = \lambda$  which implies that words of the form  $u \# \$ \in R$  – a contradiction. (Recall that we are looking for right useful words). On the other hand, the fact that  $Y = \{\$\}$  implies that  $L' = \Sigma^*$ . The second implication is proved.

From the above claim it follows that the problem QY amounts to the problem "Is  $L' = \Sigma^*$ ?". Consequently, if the former would be decidable then the latter would be also decidable for context-free languages  $L'$ .  $\square$

Notice that the solution  $Y$  in the proof of the preceding theorem has a singleton as its value. Consequently, we can use the same proof to show that for  $\diamond$  denoting controlled deletion, the problem QY' is undecidable for context-free languages  $L$  and regular languages  $R$ .

## 6 Solutions to the equation $X \diamond L = R$

We deal now with the decidability of the problem whether or not the equation  $X \diamond L = R$  has a solution  $X$ , where  $L, R$  are given languages,  $R$  regular, and  $\diamond$  is an binary language operation. The existence of a singleton solution will also be investigated. More specifically, if  $\diamond$  denotes a binary operation, given languages  $L$  and  $R$ ,  $R$  regular, the following problems will be considered:

QX: "Does there exist a solution  $X$  to the equation  $X \diamond L = R$ ?"

QX': " Does there exist a singleton solution  $Y = w$  to the equation  $Y \diamond L = R$ ?"

If  $\diamond$  denotes a controlled operation, the above questions have to be modified in a similar way as done in the preceding section.

An argument similar to that of Theorem 7 augmented by the application of Theorem 6 and the Observation after it can be used to show that the problem QX is decidable for  $\diamond$  being one of the operations: catenation, insertion, shuffle, controlled insertion, left/right quotient, deletion, scattered deletion, controlled deletion. On the other hand, a proof similar to that of Theorem 3 shows that QX' is decidable for  $\diamond$  catenation, shuffle and controlled insertion.

Let  $\diamond$  be one of the operations: catenation, insertion, shuffle, and controlled insertion. The following theorems will show that the existence of both a solution and a singleton solution  $X$  to the equation  $X \diamond L = R$  is undecidable for context-free languages  $L$  and regular languages  $R$ .

**Proposition 5** *If  $\diamond$  denotes the insertion, the problem QX is undecidable for context-free languages  $L$  and regular languages  $R$ .*

*Proof.* Let  $\Sigma$  be an alphabet with  $\text{card}(\Sigma) \geq 2$  and let  $\#$  be a letter which does not occur in  $\Sigma$ . We shall show that there exists a regular language  $R =$

$\Sigma^* \cup \{\#\}$ , such that the problem QX 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  $L'$ . For a given context-free language  $L' \subseteq \Sigma^*$  construct  $L = L' \cup \{\#\}$ .

**Claim.** For all languages  $X \subseteq \Sigma^*$  we have:

$$X \leftarrow L = R \text{ iff } X = \{\lambda\}, L = \Sigma^*,$$

where  $L, R$  are defined as above.

The implication " $\leftarrow$ " is obvious. For the reverse implication, let us assume that  $X$  contains a nonempty word  $w$ . Then the string  $w\#$  belongs to  $X \leftarrow L$  – a contradiction with the form of the words in  $R$ . Consequently, our assumption that  $X$  contains a nonempty word was false. On the other hand,  $X = \{\lambda\}$  implies  $L' = \Sigma^*$ , and the proof of the claim is complete.

The claim implies that the problem "Does there exist a language  $X$  such that  $X \leftarrow (L' \cup \{\#\}) = \Sigma^* \cup \{\#\}$ ?" amounts to the problem "Is  $L' = \Sigma^*$ ?". The theorem now follows as the latter problem is undecidable for context-free languages  $L'$ .  $\square$

Note that in the proof of the preceding theorem the language  $X = \{\lambda\}$  is a singleton. Consequently, the proof can be used to show that if  $\diamond$  denotes insertion, the problem QX' is undecidable for context-free languages  $L$  and regular languages  $R$ .

For  $\diamond$  denoting shuffle, the proof of the preceding theorem and the above remark can be used to show that, in both cases, the problems QX and QX' are undecidable for context-free languages  $L$  and regular languages  $R$ .

**Proposition 6** *If  $\diamond$  denotes the controlled insertion, the problem QX is undecidable for context-free languages  $L$  and regular languages  $R$ .*

*Proof.* Let  $\Sigma$  be an alphabet such that  $\text{card}(\Sigma) \geq 2$  and let  $b$  and  $\#$  be letters not occurring in  $\Sigma$ . There exists a regular language  $R = b\Sigma^* \cup \{b\#\}$  such that the problem QX 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  $L'$ .

For a given context-free language  $L'$ , define  $L = L' \cup \{\#\}$ .

**Claim.** For all languages  $X \subseteq (\Sigma \cup b)^*$  we have:

$$X \xleftarrow{b} L = R \text{ iff } X = \{b\} \cup X', L' = \Sigma^*,$$

where  $X' \subseteq \Sigma^*$  and  $L, R$  are defined as above.

The implication " $\Leftarrow$ " is obvious. For the reverse implication, let us assume that  $X$  contains a word  $ubv \in (\Sigma \cup b)^*b(\Sigma \cup b)^*$  different from  $b$ . Then the word  $ub\#v$  belongs to  $X \stackrel{b}{\leftarrow} L$  – a contradiction with the form of the words in  $R$ . Consequently, our assumption that such a word belongs to  $X$  was false. As the words which do not contain  $b$  do not contribute to the result, the fact that  $X$  is of the above form implies that  $L' = \Sigma^*$ . The proof of the claim is thus complete.

From the claim we deduce that the problem "Does there exist a language  $X$  such that  $X \stackrel{b}{\leftarrow} (L' \cup \{\#\}) = b\Sigma^* \cup \{b\#$ ?" amounts to the problem "Is  $L' = \Sigma^*$ ?". The theorem now follows as the latter problem is undecidable for context-free languages  $L'$ .  $\square$

A similar proof leads to the conclusion that for  $\diamond$  denoting controlled insertion,  $\text{QX}'$  is undecidable for context-free languages  $L$  and regular  $R$ . The only modification is that  $X = \{b\} \cup X'$  is replaced with  $X = \{b\}$ .

The remaining part of this section deals with problems similar to the ones studied until now, but for deletion operations. The following decidability results are basically a consequence of the fact that the result of a deletion operation from a word is a finite set.

**Theorem 8** *The problem "Does there exist a word  $w$  such that  $L \setminus w = R$ ?" is decidable for regular languages  $L$  and  $R$ .*

*Proof.* Let  $L, R$  be regular languages over an alphabet  $\Sigma$ . Notice that, if  $R$  is an infinite language, the answer to our problem is NO. If  $R$  is finite, we can effectively construct the regular set:

$$P = (LR^c)^c - \bigcup_{S \subset R} (LS^c)^c,$$

where by  $\subset$  we denote strict inclusion.

**Claim.** For all  $w \in \Sigma^*$  we have:  $w \in P$  iff  $L \setminus w = R$ .

Indeed, for given regular languages  $L$  and  $R$  we have:

$$(LR^c)^c = \{v \mid L \setminus v \subseteq R\}.$$

Therefore, if  $L \setminus w = R$  then:

$$\begin{aligned} w &\in \{v \mid L \setminus v \subseteq R\}, \\ w &\notin \{v \mid L \setminus v \subseteq S \subset R\}, \end{aligned}$$

and consequently  $w \in P$ .

For the reverse implication, let  $w$  be a word in  $P$ . As  $L \setminus w \subseteq R$  but  $L \setminus w$  is not included in any proper subset of  $R$  we have  $L \setminus w = R$ . The proof of the claim is thus complete.



The algorithm for deciding our problem will check first the finiteness of  $R$ . If  $R$  is infinite, the answer is NO. Otherwise, the set  $P$  is constructed and its emptiness is decided. If  $P = \emptyset$ , the answer is NO. Otherwise, the answer is YES and any word  $w$  in  $P$  satisfies the equation  $L \setminus w = R$ .  $\square$

The proof of Theorem 8 can be used to show that for  $\diamond$  denoting right quotient, deletion, scattered deletion and controlled deletion the problem QX' is decidable for regular languages  $L$  and  $R$ . Indeed, one only needs to replace in the preceding proof "reversed catenation" (which is the left-inverse of the left quotient) with catenation, insertion, shuffle, controlled insertion, respectively. For example, in case of deletion, the constructed set  $P$  will be:

$$P = (R^c \leftarrow L_2)^c - \bigcup_{S \subset R} (S^c \leftarrow L_2)^c.$$

The effectiveness of constructing  $P$  is based on the effectiveness of the closure of REG under the considered deletion operations (see, for example [9]).

**Theorem 9** *If  $\diamond$  denotes the iterated deletion, the problem QX' is decidable for regular languages  $L$  and  $R$ .*

*Proof.* Let  $L$  and  $R$  be regular languages over an alphabet  $\Sigma$ . If there exists a word  $w$  such that  $w \xrightarrow{*} L = R$  then  $R$  is a finite language and  $w \in R$ . Consequently, the algorithm for deciding QX' will begin by deciding the finiteness of  $R$ . If  $R$  is infinite, the answer is NO. Otherwise, for every  $w$  in  $R$  the problem of whether or not  $w \xrightarrow{*} L$  equals  $R$  is decided. (Recall that, according to the closure results from [9], the result of the iterated deletion  $w \xrightarrow{*} L$  is regular and can be effectively constructed.) If such a  $w$  is found the answer is YES, else it is NO.  $\square$

Let  $\diamond$  be a binary deletion operation. If  $L$  is a language over an alphabet  $\Sigma$ , the word  $x$  is called *left useful with respect to  $\diamond$  and  $L$*  (shortly, *useful*) if there exists a  $y \in L$  such that  $x \diamond y \neq \emptyset$ . A language  $X$  is called *left useful with respect to  $\diamond$  and  $L$*  (shortly, *left useful*), if it consists only of left useful words.

From the above definitions it follows that the problems QX, QX' are equivalent with the corresponding problems where the existence of a left useful language or word is investigated. Therefore in the sequel, when we want to prove an undecidability result about deletion, we will mean a left useful language or word when referring to a language or word whose existence is investigated.

**Proposition 7** *The problem "Does there exist a language  $X$  such that  $L \setminus X = R$ ?" is undecidable for context-free languages  $L$  and regular languages  $R$ .*

*Proof.* Let  $\Sigma$  be an alphabet,  $\text{card}(\Sigma) \geq 2$ , and let  $\#, \$_1, \$_2$  be letters which do not occur in  $\Sigma$ . There exists a singleton language  $R = \{\$_2\}$  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  $L' - L'' \neq \emptyset$ ?" for context-free languages  $L'$  and  $L''$ . For given context-free  $L', L''$ , define:

$$L = \#(L' \cup L'')\$_1 \cup \#L''\$_1\$_2.$$

**Claim.** There exists a language  $X$  such that  $L \setminus X = R$  iff  $L' - L'' \neq \emptyset$ , where  $L$  and  $R$  are defined as above.

" $\implies$ " Let  $X$  be a language such that  $L \setminus X = R$ . As  $R = \{\$_2\}$ ,  $\$_2$  has to be a suffix of all the words in  $X$ . (Recall that we are talking about left useful languages  $X$ .) Let us consider all the possible cases:

– If  $X$  contains a word of the form  $\#u\$_1\$_2\$_2$ ,  $u \in L''$ , then

$$\$_2\$_2 \in (\#u\$_1 \setminus \#u\$_1\$_2\$_2) \subseteq R.$$

– If  $X$  contains a word of the form  $\#u\$_1\$_2$ ,  $u \in L'$ , then

$$\lambda \in (\#u\$_1\$_2 \setminus \#u\$_1\$_2) \subseteq R.$$

Both possibilities lead to contradictions with the fact that  $R = \{\$_2\}$ . Consequently,  $X$  does not contain such words. Taking into account the form of the words in  $L$  and  $R$ , the only remaining possibility is that:

$$X \subseteq \{\#u\$_1\$_2 \mid u \in L' - L''\},$$

which further implies  $L' - L'' \neq \emptyset$ , since  $X$  cannot be empty.

" $\impliedby$ " Assume that  $L' - L'' \neq \emptyset$  and let  $u$  be a word in  $L' - L''$ . The language  $X = \{\#u\$_1\$_2\}$  satisfies the relation  $L \setminus X = R$ . The second implication and therefore the proof of the claim is complete.  $\square$

The proof of the preceding theorem can be used to show that for  $\diamond$  denoting right quotient,  $QX'$  is undecidable for context-free languages  $L$  and regular languages  $R$ .

Similar results are valid also for the left quotient. The languages used in the proof will be in this case:

$$R = \{\$_2\}, L = \$_1(L' \cup L'')\# \cup \$_2\$_1L''\#.$$

**Proposition 8** *If  $\diamond$  denotes the deletion, the problem  $QX'$  is undecidable for context-free languages  $L$  and regular languages  $R$ .*

*Proof.* Let  $\Sigma$  be an alphabet,  $\text{card}(\Sigma) \geq 2$ , and let  $\#_1, \#_2, \$_1, \$_2$  be letters which do not occur in  $\Sigma$ . There exists a finite language  $R = \{\#_1, \$_2\}$  such that the problem  $QX'$  is undecidable for context-free languages  $L$ . We assume the contrary and show how to solve the problem "Is  $L' \cap L'' \neq \emptyset$ ?" for context-free languages  $L', L''$ . For given context-free languages  $L', L'' \subseteq \Sigma^*$ , define the language:

$$L = \#_1\#_2L\$_1 \cup \#_2L'\$_1\$_2.$$

**Claim.** There exists a word  $w$  such that  $w \rightarrow L = R$  iff the intersection  $L' \cap L''$  is nonempty, where  $L$  and  $R$  are defined as before.

" $\Leftarrow$ " Let  $u$  be a word in  $L' \cap L''$  and take  $w = \#_1 \#_2 u \$_1 \$_2$ . The following relations hold:

$$\begin{aligned} \#_1 \#_2 u \$_1 \$_2 &\rightarrow \#_1 \#_2 u \$_1 = \{\$ _2\}, \\ \#_1 \#_2 u \$_1 \$_2 &\rightarrow \#_2 u \$_1 \$_2 = \{\#_1\}. \end{aligned}$$

Moreover, because of the markers, no other words of  $L$  are subwords of  $w$  and therefore  $w \rightarrow L = \{\#_1, \$_2\} = R$ .

" $\Rightarrow$ " Let  $w$  be a word with the property  $w \rightarrow L = R$ . As  $R = \{\#_1, \$_2\}$ , either  $\$ _2$  or  $\#_1$  is a prefix of  $w$ .

If  $\$ _2$  is a prefix of  $w$  then, necessarily,  $\#_1$  is a suffix of  $w$ , and therefore  $w$  has the form  $w = \$ _2 \alpha \#_1$ . As  $\{\$ _2\} \subseteq w \rightarrow L$ , the word  $\alpha \#_1$  has to be a subword of  $L$  – a contradiction with the form of the words in  $L$ . Consequently,  $\$ _2$  is not a prefix of  $w$ .

If  $\#_1$  is a prefix of  $w$  then, necessarily,  $\$ _2$  is a suffix of  $w$ , and therefore  $w$  has to be of the form  $w = \#_1 \alpha \$ _2$ . As  $\{\$ _2\} \subseteq w \rightarrow L$  it results that  $w$  has the form  $w = \#_1 \#_2 u \$_1 \$_2$  where  $u$  belongs to  $L'$ . As  $\{\#_1\} \subseteq w \rightarrow L$  it results that  $w$  has the form  $w = \#_1 \#_2 u' \$_1 \$_2$ , where  $u'$  belongs to  $L''$ . We conclude that  $u = u' \in L' \cap L''$ , which is therefore nonempty. The proof of the claim is complete.

The claim implies that the problem "Does there exist a word  $w$  such that:

$$w \rightarrow (\#_1 \#_2 L \$_1 \cup \#_2 L' \$_1 \$_2) = \{\#_1, \$_2\}?"$$

amounts to the problem "Is  $L' \cap L'' = \emptyset$ ?" □

*Remark.* The operations of insertion and deletion are associated with several notions rather basic in the combinatorics of words. While a more detailed study of such notions lies outside the scope of this paper, we want to briefly mention here one of them, which has been studied in [12], [13].

A language  $L'$  is called a *deletion set* if

$$L' = w \rightarrow L'',$$

for some word  $w$  and language  $L''$ . Clearly, every deletion set is finite. It is also not difficult to prove that it is decidable whether or not a given finite language is a deletion set. This result should be contrasted with the undecidability result of Proposition 8.

**Proposition 9** *If  $\diamond$  denotes the deletion the problem  $QX$  is undecidable for context-free languages  $L$  and regular languages  $R$ .*

*Proof.* Let  $\Sigma$  be an alphabet,  $\text{card}(\Sigma) \geq 2$ , and let  $\$, \$_1, \$_2, \#$  be letters which do not occur in  $\Sigma$ . There exists a singleton language  $R = \{\$\}$  such that the problem QX is undecidable for context-free languages  $L$ . We assume the contrary and show how to solve the problem "Is  $L' - L'' = \emptyset$ ?" for context-free languages  $L'$  and  $L''$ . Let  $L', L''$  be context-free languages and consider the language:

$$L = \#(L' \cup L'')\$_1 \cup \#L''\$_1\$_2 \cup \$_2\#L''\$_1.$$

**Claim.** There exists a language  $X$  such that  $X \rightarrow L = R$  iff  $L' - L'' \neq \emptyset$ , where  $L$  and  $R$  are defined as before.

" $\implies$ " Let  $X$  be a language such that  $X \rightarrow L = R$ . Every word in  $X$  has to be of the form  $w\$$  or  $\$w$  where  $w$  is a word in  $L$ . (Recall that we are considering only left useful languages  $X$ .) For a word belonging to  $X$  there are two possibilities: it either consists of a word of  $L''$  bounded by markers or of a word of  $L' - L''$  bounded by markers. If the first situation holds, a word belonging to  $X$  has one of the following forms:  $\#u\$_1\$_2\$_2$ ,  $\$_2\#u\$_1\$_2$ ,  $\$_2\$_2\#u\$_1$ ,  $\#u\$_1\$_2$ ,  $\$_2\#u\$_1$ ,  $u \in L''$ . All the mentioned cases lead to contradictions. For example, if  $\$_2\#u\$_1$ ,  $u \in L''$  belongs to  $X$ , then

$$\lambda \in (\$_2\#u\$_1 \rightarrow \$_2\#L''\$_1) \subseteq R,$$

which contradicts the fact that  $R = \{\$\}$ . Consequently,  $X$  does not contain such words and the first situation cannot happen.

If the second situation occurs then, taking into account the form of the words in  $L$  and  $R$ , we conclude that:

$$X \subseteq \{\#u\$_1\$_2 \mid u \in L' - L''\} \cup \{\$_2\#u\$_1 \mid u \in L' - L''\}.$$

This further implies that if  $X$  with the desired property exists then  $L' - L'' \neq \emptyset$ .

" $\impliedby$ " Assume that  $L' - L'' \neq \emptyset$  and let  $u$  be a word in  $L' - L''$ . The language  $X = \{\#u\$_1\$_2\}$  satisfies the relation  $X \rightarrow L = R$ . The proof of the claim is thus complete.  $\square$

**Proposition 10** *If  $\diamond$  denotes the controlled deletion, the problem QX is undecidable for context-free languages  $L$  and regular languages  $R$ .*

*Proof.* Let  $\Sigma$  be a language,  $\text{card}(\Sigma) \geq 2$ , and let  $\#, \#_1, \#_2, \$_1, \$_2$  be letters which do not occur in  $\Sigma$ . There exists a singleton language  $R = \{\#_1\$_2\}$  such that the problem QX is undecidable for context-free languages  $L$ . We assume the contrary and show how to solve the problem "Is  $L' - L'' = \emptyset$ ?" for context-free languages  $L', L''$ .

For given  $L', L''$  as before consider the language  $L$  defined by:

$$L = \#_2(L' \cup L'')\$_1 \cup \#_2L'\$_1\$_2.$$

**Claim.** There exists a language  $X$  such that  $X \xrightarrow{\#_1} L = R$ , iff  $L' - L'' \neq \emptyset$ , where  $L, R$  are defined as before.

" $\implies$ " If  $X$  would contain a word of the form  $\#_1\#_2u\$_1\$_2\$_2$ ,  $u \in L''$ , then

$$\#_1\$_2\$_2 \in (\#_1\#_2u\$_1\$_2\$_2 \xrightarrow{\#_1} L) \subseteq R,$$

– a contradiction.

If  $X$  would contain a word of the form  $\#_1\#_2u\$_1\$_2$ ,  $u \in L''$ , then

$$\#_1 \in (\#_1\#_2u\$_1\$_2 \xrightarrow{\#_1} L) \subseteq R,$$

– a contradiction.

Consequently, the only possibility that remains is that

$$X \subseteq \{\#_1\#_2u\$_1\$_2 \mid u \in L' - L''\},$$

and, as  $X \neq \emptyset$ , this implies  $L' - L'' \neq \emptyset$ .

" $\impliedby$ " If  $u$  is a word in  $L' - L''$  take  $X = \{\#_1\#_2u\$_1\$_2\}$ . The language  $X$  satisfies the equality  $X \xrightarrow{\#_1} L = R$ . The proof of the claim and therefore of the theorem is complete.  $\square$

The previous proof can be used to show that for  $\diamond$  denoting controlled deletion the problem  $QX'$  is undecidable for context-free languages  $L$  and regular languages  $R$ .

**Proposition 11** *If  $\diamond$  denotes the scattered deletion, the problems  $QX$ ,  $QX'$  are undecidable for context-sensitive languages  $L$  and regular languages  $R$ .*

*Proof.* We shall prove a stronger result: there exists a singleton language,  $R = \{\lambda\}$ , such that the problem  $QX$  is undecidable for context-sensitive languages  $L$ . We assume the contrary and show how to solve the emptiness problem for context-sensitive languages. This follows noticing that the problem "Does there exist a language  $X$  such that  $X \rightsquigarrow L = \{\lambda\}$ ?" is equivalent with the problem "Is  $L \neq \emptyset$ ?". Indeed, if  $L \neq \emptyset$  we can take  $X = \{w\}$ , where  $w$  is one of the shortest words in  $L$ . It is easy to see that  $\{w\} \rightsquigarrow L = \{\lambda\}$ . The reverse implication is obvious.  $\square$

## 7 Conclusions and open problems

Theorems 5 and 6 prove to be a powerful tool for investigating equations of the form  $L \diamond X = R$  (respectively  $X \diamond L = R$ ) in case the operation  $\diamond$  possesses a right-inverse (respectively a left-inverse). They provide the biggest language  $R'$  with the property  $L \diamond R' \subseteq R$  (respectively  $R' \diamond L \subseteq R$ ). Consequently, if a

solution to the equation exists, the language  $R'$  will also be a solution, namely the maximal one.

These results can also be used for finding solutions to more general "linear" equations such as:

$$(L_1 \diamond_1 X) \diamond_2 L_2 = R,$$

$$(L_1 \diamond_1 X) \cup (L_2 \diamond_2 X) = R,$$

to "linear" systems of equations:

$$\begin{aligned} (L_1 \diamond_1 X) \cup (L_2 \diamond_2 Y) &= R_1 \\ (L_3 \diamond_3 X) \cup (L_4 \diamond_4 Y) &= R_2, \end{aligned}$$

or to quadratic equations such as

$$X^2 = R \text{ or } X \diamond X = R \text{ and } L \diamond X \diamond X = R.$$

The problem of whether the existence of solutions to the equations  $L \diamond Y = R$ ,  $X \diamond L = R$  is decidable for given regular  $L$  and  $R$  remains open for  $\diamond$  denoting iterated insertion, permuted insertion, permuted scattered insertion, iterated deletion, permuted deletion, permuted scattered deletion. The difficulty arises from the fact that REG is either not closed under the considered operation, or is not closed under its righ-(respectively left) inverse.

Is is also an open problem whether the existence of a solution to  $X \diamond L = R$  is undecidable for given context-free  $L$  and regular  $R$ , in case of  $\diamond$  denoting iterated insertion, permuted insertion, permuted scattered insertion, iterated deletion, permuted deletion, permuted scattered deletion.

One obvious direction of research would be the study of the existence of solutions to equations  $L \diamond X = R$  for context-free or context-sensitive languages  $R$  or the study of equations of higher degree.

*Acknowledgements* We would like to thank to Professor Jean-Pierre Olivier for extended discussions and for pointing out the references in [3], [4]. The valuable suggestions of Professor Juhani Karhumäki and Dr. Jarkko Kari are gratefully acknowledged.

## References

- [1] R.Book, M.Jantzen, C.Wrathall. Monadic Thue systems. *Theoretical Computer Science*, 19 (1982), pp.231-251.
- [2] R.Book, M.Jantzen, C,Wrathall. Erasing strings. *Lecture Notes in Computer Science*, 104 (1981), pp.252-260.
- [3] J.H.Conway. *Regular Algebra and Finite Machines*. Chapman and Hall Mathematics Series, London, 1971.

- [4] P.J.Freyd, A.Scedrov. *Categories, Allegories*. North-Holland (1990).
- [5] S.Ginsburg. *Algebraic and Automata-Theoretic Properties of Formal Languages*. North-Holland, Amsterdam, 1975.
- [6] M.Jantzen. Confluent string rewriting and congruences. *EATCS Bulletin*, 28 (1986), pp.52-72.
- [7] M.Jantzen. Semi-Thue systems and generalized Church-Rosser properties. *Proc. Fete des Mots*, Rouen, France, 1982, 60-75.
- [8] J.Karhumäki. Equations over finite sets of words and equivalence problems in automata theory. *Words, Languages and Combinatorics*, World Scientific Publishing, 1990.
- [9] L.Kari. On insertion and deletion in formal languages. *Ph.D. Thesis*, University of Turku, 1991.
- [10] L.Kari. Insertion and deletion of words: determinism and reversibility. *Lecture Notes in Computer Science*, vol.629, 1992, pp.315-327.
- [11] L.Kari. Generalized derivatives. *Fundamenta Informaticae*, vol.18, nr.1, 1993, pp.27-40.
- [12] L.Kari, A.Mateescu, G.Paun, A.Salomaa. Deletion sets. To appear in *Fundamenta Informaticae*.
- [13] L.Kari, A.Mateescu, G.Paun, A.Salomaa. On parallel deletions applied to a word. Submitted.
- [14] W.Kuich, A.Salomaa. *Semirings, Automata, Languages*. Springer Verlag, Berlin, 1986.
- [15] A.Salomaa. *Formal Languages*. Academic Press, London, 1973.
- [16] A.Salomaa, M.Soittola. *Automata-Theoretic Aspects of Formal Series*, Springer Verlag, Berlin, Heidelberg, New York, 1978.
- [17] L.Sântean. Six arithmetic-like operations on languages. *Revue Roumaine de Linguistique*, Tome XXXIII, 1988, Cahiers de linguistique theorique et applique, Tome XXV, 1988, No.1, Janvier-Juin, pp.65-73.