Invertible insertion and deletion operations

Lila Kari
Academy of Finland and Department of Mathematics[1]
University of Turku
20500 Turku
Finland

Abstract

The paper investigates the way in which the property of a language operation $\diamond$ "to be invertible" helps in solving language equations of the type $L \diamond Y = R$. In the beginning, the simple case where $\diamond$ denotes catenation is studied, but the results are then generalized for various invertible insertion and deletion operations. For most of the considered operations $\diamond$, the problem "Does there exist a solution Y to the equation $L \diamond Y = R$?" turns out to be decidable for given regular languages L and R.

1 Introduction

Starting with the definition of a context-free language as the minimal solution of a system of equations (see [13]), continuing with equations and systems of equations over free monoids or free semigroups (see [5] and its references), language and word equations have constantly played a central role in formal language theory. In this paper we study language equations of the form $L \diamond Y = R$, where $\diamond$ is an invertible 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 [2]. 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 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

[1]The work reported here is part of the project 11281 of the Academy of Finland

catenation and quotient have been investigated in [6], [14]. Some of them are mentioned in Section 3 and among them we can list 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 which possess some kind of *inverses* bears some resemblance to questions studied for categories of abstract binary relations (see, for instance, [3]).

One of the results states that, if $\diamond$ is a binary operation possessing a "right-inverse " and a solution to the equation exists, then a maximal solution R' can also be found. The language 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. Almost all the considered operations $\diamond$ are non-commutative. Section 5 deals with the analogous equation $X \diamond L = R$ and its solutions, where $\diamond$ is an operation possessing a "left-inverse".

Section 6 investigates the decidability of the existence of solutions to the above equations. For most operations, the problem turns out to be decidable for L and R given regular languages.

Section 7 points out how the obtained results can be used to solve more general linear equations and systems of equations, as well as quadratic and other equations.

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

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

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

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 $\mathrm{Mi}(u)$. For two languages L_1 and L_2 over Σ^*,

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

REG denotes the family of regular languages. For unexplained formal language notions the reader is referred to [12].

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 Σ, 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:

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

(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 $v = (u\backslash uv) \subseteq (L\backslash R^c)$ - 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 $L\backslash R^c$, there exist words $w \in R^c$, $u \in L$, such that $uv = w$. 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. ♡

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 Σ define:

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

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. ♡

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 Σ 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 $\lg(w) \leq m$. The answer is YES if such a word w is found, and NO otherwise. $\heartsuit$

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 Σ be an alphabet, $\mathrm{card}(\Sigma) \geq 2$, and let $\#$ be a letter which does not occur in Σ. 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 Σ 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. $\heartsuit$

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 Σ. 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. $\heartsuit$

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 – 4, Proposition 1 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 $R''_1, R''_2, \ldots, R''_m$, $m \geq 1$ 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 $R''_1, \ldots, R''_m$ 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, we can prove results similar to the ones in Section 2 also for more general invertible word operations. Actually, as Theorem 5 will show, one can generalize Theorem 1 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 invertible binary word operations for which theorems similar to Theorem 1 hold. For a more detailed study of these operations, see [6], [14]. 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:

$$u \longleftarrow v = \{u_1 v u_2 | \; u = u_1 u_2, \; u_1, u_2 \in \Sigma^*\}.$$

For example, $cd \longleftarrow a = \{acd, cad, cda\}$, where a, c, d are letters in Σ. 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 [4] for details).

A more exotic variant of insertion is obtained if we combine the ordinary insertion with the commutative variant. The commutative variant 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:

$$u \leftsquigarrow v = u \longleftarrow \mathrm{com}(v).$$

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

$$u \stackrel{a}{\longleftarrow} v = \{u_1avu_2 |\ u = u_1au_2\}.$$

Special cases of catenation can be now obtained by using a marker and the 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 [11]):

$$u \amalg v = \{u_1v_1u_2v_2 \ldots u_kv_k |\ u = u_1 \ldots u_k, v = v_1 \ldots v_k, u_i, v_i \in \Sigma^*, k \geq 1\}.$$

Otherwise, the *permuted scattered insertion* is obtained:

$$u \leftsquigarrow v = u \amalg \mathrm{com}(v).$$

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:

$$u \longrightarrow v = \{w |\ u = w_1vw_2, w = w_1w_2\}.$$

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] 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 [7], [8], [9], [10].

The *permuted deletion* of v from u is

$$u \rightsquigarrow v = u \longrightarrow \mathrm{com}(v).$$

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

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

The *scattered deletion* of v from u is

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

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

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:

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

4 Operations possessing right-inverses

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:

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

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

$$w \in (u \diamond v) \textit{ iff } v \in (u \Box w).$$

In other words, the operation $\Box$ 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 $\Box$. 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 Σ and $\diamond$, $\Box$ 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 $R' = (L \Box R^c)^c$ 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 $\Box$ is the right-inverse of $\diamond$, we further deduce that v belongs to $(u \Box w)$ which is a subset of $L \Box R^c$. 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 $R'^c = L \Box R^c$, there exist words $w \in R^c$, $u \in L$ such that $v \in (u \Box w)$. As $\Box$ 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. ♡

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 us a powerful tool for solving the equation $L \diamond Y = R$, when L and R are regular languages. The following Proposition will allow us to apply Theorem 5 to the 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 $\diamond$.*

Proposition 2 *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.

Proof. Let Σ be an alphabet and u, v, w words over the alphabet Σ.

Catenation. The word w equals uv iff $v = u\backslash w$.

Insertion. The word w belongs to $(u \longleftarrow v)$ iff $w = u_1 v u_2$, $u = u_1 u_2$, which happens exactly in case $v \in (w \rightleftharpoons u)$.

Shuffle. The word w belongs to $(u \amalg v)$ iff $w = u_1 v_1 \ldots u_k v_k$, $u_i, v_i \in \Sigma^*$, $1 \leq i \leq k$, $u = u_1 \ldots u_k$, $v = v_1 \ldots v_k$. This, in turn holds iff v belongs to $w \dashrightarrow u$.

Right quotient. The word w equals u/v iff $u = wv$ iff $v \in w\backslash u$.

Deletion. The word w belongs to $u \longrightarrow v$ iff $u = w_1 v w_2$, $w = w_1 w_2$. This happens exactly in case $v \in (u \rightleftharpoons w)$.

Scattered deletion. The word w belongs to $u \dashrightarrow v$ iff $u = w_1 v_1 \ldots w_k v_k$, $w_i, v_i \in \Sigma^*$, $1 \leq i \leq k$ and $v = v_1 \ldots v_k$, $w = w_1 \ldots w_k$. This, in turn, holds iff v belongs to $u \dashrightarrow w$.

Controlled insertion. Define the gsm:

$$\begin{aligned} g_a = \; & (\Sigma, \Sigma \cup \{\#, \$\}, \{s_0, s, s'\}, s_0, \{s'\}, P_a), \\ P_a = \; & \{s_0 b \longrightarrow b s_0 | \; b \in \Sigma\} \cup \{s_0 a \longrightarrow a\#\$s\} \cup \\ & \{s_0 a \longrightarrow a\#s\} \cup \{sb \longrightarrow bs | \; b \in \Sigma\} \cup \\ & \{sb \longrightarrow b\$s' | \; b \in \Sigma\} \cup \{s'b \longrightarrow bs' | \; b \in \Sigma\}, \end{aligned}$$

where $\#, \$$ are new symbols which do not occur in Σ. It is easy to prove that for every language $L \subseteq \Sigma^*$ we have:

$$g_a(L) = \{ua\#w\$v | \; u, v, w \in \Sigma^*, \text{ and } uawv \in L\}. \qquad (*)$$

Define the morphism $h : (\Sigma \cup \{\#, \$\})^* \longrightarrow \Sigma^*$ by:

$$h(\#) = h(\$) = \lambda, \; h(a) = a, \forall a \in \Sigma.$$

Define now the binary word operation:

$$u \Box v = h((g_a(u) \rightleftharpoons v) \cap \#\Sigma^* \$).$$

The operation $\Box$ is the right-inverse of controlled insertion next to the letter a. Indeed, the word w belongs to $u \overset{a}{\longmapsfrom} v$ iff $w = u_1 a v u_2$, $u = u_1 a u_2$. This, in turn, happens iff v is an element of the set $h((g_a(w) \rightleftharpoons u) \cap \#\Sigma^*\$) = w \Box u$.

Controlled deletion. Consider the gsm and morphism defined above. The operation defined by

$$u \Box v = h((g_a(v) \rightleftharpoons u) \cap \#\Sigma^*\$)$$

is the right-inverse of controlled deletion. Indeed, w belongs to $(u \overset{a}{\longmapsto} v)$ iff $u = w_1 a v w_2$, $w = w_1 a w_2$. This happens exactly in case v is in $u \Box w$.

Permuted insertion. The word w belongs to the set $u \leftsquigarrow v$ iff $w = u_1 v' u_2$, $v \in \text{com}(v')$. This happens exactly in case $v \in \text{com}(w \rightleftharpoons u)$.

Permuted scattered insertion. The word w is an element of the set $u \leftsquigarrow v$ iff $w = u_1 v_1 \ldots u_k v_k$, $u_i, v_i \in \Sigma^*$, $1 \leq i \leq k$ and $u = u_1 \ldots u_k$, $v \in \text{com}(v_1 \ldots v_k)$. This, in turn, holds iff $v \in \text{com}(w \dashrightarrow u)$.

Permuted deletion. The word w belongs to $u \rightsquigarrow v$ iff $u = w_1 v' w_2$, $w = w_1 w_2$, $v' \in \text{com}(v)$. This holds iff $v \in \text{com}(u \rightleftharpoons w)$.

Permuted scattered deletion. The word w belongs to the set $u \rightsquigarrow v$ iff $u = w_1 v_1 \ldots w_k v_k$, $w_i, v_i \in \Sigma^*$, $1 \leq i \leq k$, and $v \in \text{com}(v_1 \ldots v_k)$, $w = w_1 \ldots w_k$. This happens exactly in case $v \in \text{com}(u \dashrightarrow w)$. $\heartsuit$

Together with Theorem 5, Proposition 2 allows us to investigate the solutions to the equation $L \diamond Y = R$ for regular languages L, R. Indeed, one can prove theorems analogous to Theorem 1 for the following operations: insertion, shuffle, left quotient, right quotient, deletion, scattered deletion, dipolar deletion, controlled insertion, controlled deletion (see the closure properties proved in [6]).

5 Operations possessing left-inverses

As we have seen in Section 2, the results concerning the equation $LY = R$ could be transferred without much difficulty to the equation $XL = R$. This naturally gives the idea that it is possible to obtain general results as Theorem 5 also for equations $X \diamond L = R$. With this in mind, a notion corresponding to that of right-inverse has to be defined.

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

$$w \in (u \diamond v) \textit{ iff } u \in (w \Box v).$$

In other words, the operation $\Box$ 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 $\Box$. The relation "is the left-inverse of" is symmetric.

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

Using the notion of left-inverse, we are now ready to state a twin theorem of Theorem 5, for the equation $X \diamond L = R$.

Theorem 6 *Let L, R be languages over an alphabet Σ and $\diamond$, $\Box$ 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 $R' = (R^c \Box L)^c$ 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 $\Box$ with $\Box^r$. ♡

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 proposition will allow us to investigate the solutions of the equation $X \diamond L = R$ for the operations defined in Section 3.

Proposition 3 *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.*

Proof. Let Σ be an alphabet and u, v, w be words in Σ.

Catenation. The word w equals uv iff $u \in w/v$.

Insertion. The word w belongs to $u \longleftarrow v$ iff $w = u_1 v u_2$, $u = u_1 u_2$, which happens exactly in case u is in $w \longrightarrow v$.

Controlled insertion. The word w belongs to $u \overset{a}{\longleftarrow\!\!\!|}\, v$ iff $w = u_1 a v u_2$, $u = u_1 a u_2$ iff $u \in (w \overset{a}{\longmapsto} v)$.

Shuffle. The word w belongs to $u \amalg v$ iff $w = u_1 v_1 \ldots u_k v_k$, $u_i, v_i \in \Sigma^*$, $1 \le i \le k$ and $u = u_1 \ldots u_k$, $v = v_1 \ldots v_k$. This, in turn, holds iff u belongs to $w \dashrightarrow v$.

Permuted insertion. The word w is in the set $u \leftsquigarrow v$ iff $w = u_1 v' u_2$, $u = u_1 u_2$, $v' \in \mathrm{com}(v)$. This holds iff $u \in (w \rightsquigarrow v)$.

Permuted scattered insertion. The word w belongs to $u \leftsquigarrow v$ iff we have $w = u_1 v_1 \ldots u_k v_k$, $u_i, v_i \in \Sigma^*$, $1 \le i \le k$, and $u = u_1 \ldots u_k$, $v \in \mathrm{com}(v_1 \ldots v_k)$. This happens exactly in case $u \in (w \rightsquigarrow v)$.

Left quotient. The word w equals $u \backslash v$ iff $u = vw$.

Dipolar deletion. The word w belongs to the set $u \rightleftharpoons v$ iff $u = v_1 w v_2$, $v = v_1 v_2$. This, in turn, happens exactly in case u is in the set $v \longleftarrow w$. ♡

Together with Theorem 6, the preceding Proposition allows us to solve to the equation $X \diamond L = R$, when L, R are regular, for the following operations: insertion, controlled insertion, shuffle, right/left quotient, deletion, scattered deletion, controlled deletion (see [6] for the closure properties of REG under these operations).

6 Decidability issues

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

The problem turns out to be decidable in case REG is closed under the operation $\diamond$ and its inverse. 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. Then the problem "Does there exist a solution Y to the equation $L \diamond Y = R$ (respectively a solution X to the equation $X \diamond L = R$) is decidable for regular languages L and R.*

Proof. Analogous to that of Theorem 2 and using the results from Theorem 5, Theorem 6 and the closure properties of REG under the above operations (see [6]) which are all constructive. ♡

Let $\diamond$ denote one of the operations: catenation, shuffle, permuted insertion, permuted scattered insertion, controlled insertion. A proof similar to that of Theorem 3 can be used to show that in all mentioned cases the existence of a singleton solution to the equation $L \diamond Y = R$ is decidable for regular languages L and R. In the same way one can show that the existence of a singleton solution to the equation $X \diamond L = R$ is decidable for $\diamond$ denoting shuffle and controlled insertion.

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 the existence of both a solution and a singleton solution to $L \diamond Y = R$ is undecidable for context-free languages L and regular languages R. (If $\diamond$ stands for controlled insertion we choose the control letter to be $\#$.)

The folowing decidability results are basically due to 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 $L \backslash w = R$?" is decidable for regular languages L and R.*

Proof. Let L, R be regular languages over an alphabet Σ. 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:

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

where by $\subset$ we denote strict inclusion.

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

Indeed for given regular languages L and R we have:

$$(LR^c)^c = \{v | \ L \backslash v \subseteq R\}.$$

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

$$\begin{array}{ll} w \in & \{v | \ L \backslash v \subseteq R\}, \\ w \notin & \{v | \ L \backslash v \subseteq S \subset R\}, \end{array}$$

and consequently $w \in P$.

For the reverse implication, let w be a word in P. As $L \backslash w \subseteq R$ but $L \backslash w$ is not included in any proper subset of R we have $L \backslash w = R$. 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. Else, the set P is constructed and its emptiness is decided. If $P = \emptyset$, the answer is NO. Else the answer is YES and any word w in P satisfies the equation $L \backslash w = R$. ♡

The proofs of the preceding theorem can be used to show that for $\diamond$ denoting right quotient, deletion, scattered deletion and controlled deletion, the existence of a singleton solution to the equation $X \diamond L = R$ 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 the case of deletion, the constructed set P will be:

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

The effectiveness of constructing P is based on the effectiveness of the closure of REG under the considered operations (see, for example, [6]).

Theorem 9 *If $\diamond$ denotes the iterated deletion, the problem "Does there exist a singleton solution to the equation $X \diamond L = R$?" is decidable for regular languages L and R.*

Proof. Let L and R be regular languages over an alphabet Σ. If there exists a word w such that $w \longrightarrow^* L = R$ then R is a finite language and $w \in R$. Consequently, the algorithm for deciding our problem will begin by deciding the finiteness of R. If R is infinite, the answer is NO. Else, for every w in R the problem of whether or not $w \longrightarrow^* L$ equals R is decided. (Recall that, according to the closure results from [6], the result of the iterated sequential deletion $w \longrightarrow^* L$ is regular and can be effectively constructed.) If such a w is found the answer is YES, else it is NO. ♡

7 Conclusions and open problems

Theorems 5 and 6 prove to be a powerful tool for investigating equations of the form $L \diamond Y = R$ (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 $R' \diamond L \subset R$). 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 whether the existence of solutions to the equations $L \diamond Y = R$, $X \diamond L = R$ is decidable for given regular languages 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 inverse.

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

Acknowledgements We would like to thank Professor Jean-Pierre Olivier for extended discussions and for pointing out the references in [2], [3]. The valuable suggestions of Professor Juhani Karhumäki and Dr. Jarkko Kari are gratefully acknowledged.

References

[1] R.V.Book, M.Jantzen, C.Wrathall. Monadic Thue systems. *Theoretical Computer Science*, **19** (1982), 231-251.

[2] J.H.Conway. *Regular Algebra and Finite Machines*. Chapman and Hall Mathematics Series, London, 1971.

[3] P.J.Freyd, A.Scedrov. *Categories, Allegories*. North-Holland, 1990.

[4] M.Jantzen. Semi-Thue systems and generalized Church-Rosser properties. *Proc. Fete des Mots*, Rouen, France, 1982, 60-75.

[5] J.Karhumäki. Equations over finite sets of words and equivalence problems in automata theory. *Words, Languages and Combinatorics*, World Scientific Publishing, 1990.

[6] L.Kari. On insertion and deletion in formal languages. *Ph.D. Thesis*, University of Turku, 1991.

[7] L.Kari. Insertion and deletion of words: determinism and reversibility. *Lecture Notes in Computer Science*, vol.629, 1992, pp.315-327.

[8] L.Kari. Generalized derivatives. *Fundamenta Informaticae*, vol.18, nr.1, 1993, pp.27-40.

[9] L.Kari, A.Mateescu, Gh.Paun, A.Salomaa. Deletion sets. To appear in *Fundamenta Informaticae*.

[10] L.Kari, A.Mateescu, Gh.Paun, A.Salomaa. On parallel deletions applied to a word. To appear in RAIRO.

[11] W.Kuich, A.Salomaa. *Semirings, Automata, Languages*. Springer Verlag, Berlin, 1986.

[12] A.Salomaa. *Formal Languages*. Academic Press, London, 1973.

[13] A.Salomaa, M.Soittola. *Automata-Theoretic Aspects of Formal Series*, Springer Verlag, Berlin, Heidelberg, New York, 1978.

[14] L.Santean. Six arithmetic-like operations on languages. *Rev. Roumaine de Linguistique*, XXXIII(1988), Cah. de ling. theor. et appl. 1, XXV(1988).