Semi-Commutativity Sets of Morphisms over Finitely Generated Free Monoids[1]

Lila Kari[2], Gheorghe Păun[3], Arto Salomaa[2]

June 7, 2010

Abstract

The notion of a semi-commutativity set for word mappings was defined in [3] as an abstraction of a problem in cryptography. The notion is of special interest in case the mappings are morphisms. Then rather surprising constructions become possible. We investigate such constructions, paying special attention to exceptional values of inverse mappings. Some of our results bear a close relation to certain important issues in the theory of formal languages.

1 Background of the problem

Consider the following problem in the theory of cryptographic protocols. A seller, S, possesses a number of secrets. S has published a list of descriptions of the secrets and is offering them for sale at a price that is the same for each of the secrets. A buyer B wants to buy one of the secrets. However, he is not willing to disclose to anybody, not even to S, which of the secrets he wants. On the other hand, B should not learn more than one secret.

These two seemingly contradictory requirements, S disclosing the secret B wants but no other secrets and S not knowing which secret he disclosed, can be fulfilled using cryptographic protocols based on one-way functions. The first solution suggested in [1] is very complicated. The solution in [7] is simple but requires several buyers. Recently a simple solution in the original set-up of one buyer has been given in [4]. The protocol for secret selling of secrets can also be used as a building block for more complicated protocols, [2]. The reader is referred to [6] as a general introduction to one-way functions and cryptographic protocols.

[1]The work reported here has been supported by the Project 11281 of the Academy of Finland

[2]Academy of Finland and Mathematics Department, University of Turku, 20500 Turku, Finland

[3]Institute of Mathematics,Str.Academiei 14, 70109 Bucuresti, Romania

Let us look in more detail into our problem. Consider the secrets as words

$$w_1, w_2, \ldots, w_n.$$

More explicitly, the index i in w_i tells what the secret is about, that is, proposes a question, whereas the word w_i gives the answer. The seller S has his own personal encryption and decryption functions E_S and D_S such that

$$D_S(E_S(w)) = w,$$

for all values w under consideration. Similarly, B has his functions E_B and D_B. We can now define a protocol as follows.

Step 1. S gives B the sequence

$$E_S(w_1), \ldots, E_S(w_n).$$

Step 2. If B wants the secret indexed by b, he gives S the value $E_B E_S(w_b)$.

Step 3. S gives B the value $D_S E_B E_S(w_b)$. □

If we now have

$$(*) \qquad D_B D_S E_B E_S(w_b) = w_b,$$

B has learned the secret he wants. There are further cryptographic requirements.

(i) B should not be able to learn anything from the words he gets in Step 1. In particular, he should not be able to determine the mapping D_S, although he knows $x = E_B E_S(w_b)$ and $D_S(x)$.

(ii) Although S knows both all words w_i and $E_S(w_i)$, he should not be able to learn b from $E_B E_S(w_b)$.

The requirements (i) and (ii) concern the specific encryption and decryption methods used, and will not be dealt with in this paper. Instead, we focus the attention on the equation $(*)$. Thus, cryptographic properties of the E's and D's will be ignored. Essential in our considerations will be that D_S and D_B are left inverses of E_S and E_B, respectively. Such a study of the equation $(*)$ was begun in [3].

Of special interest is the case where the mappings E_S and E_B are morphisms in the language–theoretic sense. Then the resulting problems concern also some rather basic issues of language theory. This will be the subject matter of the present paper.

The basic definitions and some initial observations will be given in the next section. Section 3 solves some problems posed in Section 2 and, at the same time, exhibits some constructions dealing with inverses. A very crucial issue is how the values may be defined for left inverses in case the values are not "forced" by the argument values. This leads to three classes – smooth, semi–smooth and unrestricted–left inverses investigated in Sections 4 – 6. Undecidability results are presented in Section 7 for the unrestricted case. Finally, we outline some further work and open problems.

We refer to [5] for all unexplained notions in language theory, in particular, for issues dealing with equality sets and codes.

2 Definitions and the basic set-up

Let Σ be a finite alphabet and Σ^* the set of all words over Σ, including the empty word λ. In algebraic terms, Σ^* is the free monoid generated by Σ, where the operation is catenation (juxtaposition) and λ is the unit element. The length of a word $w \in \Sigma^*$ is denoted by $|w|$. We will be mostly concerned with *morphisms* (usually denoted by f and g) of Σ^* into Σ^*. Clearly, such a morphism f is completely defined if the values $f(a)$, for $a \in \Sigma$, are specified.

Our starting point is the requirement $(*)$, now written in the form

$$(*)' \qquad f^{-1}g^{-1}fg(w) = w, \text{ for all } w \in \Sigma^*.$$

Here f^{-1} and g^{-1} are *left inverses* of f and g, that is,

$$(**) \qquad f^{-1}f(w) = w = g^{-1}g(w), \text{ for all } w \in \Sigma^*.$$

The existence of left inverses satisfying $(**)$ implies that f and g are injective. According to the customary terminology, [5], injective morphisms over Σ^* are called *codes*.

The condition $(*)'$ was investigated in [3] in the case, where f and g are arbitrary word mappings, or even mappings defined on an arbitrary set rather than Σ^*. The case studied in this paper, where f and g are codes, is of special interest in language theory because some fundamental issues such as equality sets are involved.

We now proceed to the basic definitions. The *equality* and *commutativity sets* for pairs (f, g) of morphisms are defined by:

$$E(f, g) = \{w \in \Sigma^* |\ f(w) = g(w)\},$$

$$\mathrm{COM}(f, g) = \{w \in \Sigma^* |\ fg(w) = gf(w)\}.$$

The reader is referred to [5] for the basic theory concerning equality sets. For commutativity sets, it is important that also the range alphabet of the morphisms equals Σ. Clearly,

$$\mathrm{COM}(f, g) = E(fg,\ gf).$$

Let f and g be codes and let f^{-1} and g^{-1} be (some of) their left inverses. Then the *semi–commutativity set* associated to the ordered quadruple (f^{-1}, g^{-1}, f, g) is defined by

$$\mathrm{SCOM}(f^{-1}, g^{-1}, f, g) = \{w \in \Sigma^* |\ f^{-1}g^{-1}fg(w) = w\}.$$

The ordered quadruple (f^{-1}, g^{-1}, f, g) is termed *semi–commutative* iff

$$\mathrm{SCOM}(f^{-1}, g^{-1}, f, g) = \Sigma^*.$$

We will see that the order of f and g is important in the definition: a different semi–commutativity set may result if the order of f and g is changed. Semi-commutativity sets can sometimes be expressed in terms of equality sets as follows:

$$\mathrm{SCOM}(f^{-1}, g^{-1}, f, g) = E(f^{-1}g^{-1}fg, id),$$

where id is the identity morphism. This representation is not always valid because, as will be seen below, $f^{-1}g^{-1}fg$ is not always a morphism. It is also clear that the commutativity set $\mathrm{COM}(f, g)$ is included in the intersection

$$\mathrm{SCOM}(f^{-1}, g^{-1}, f, g) \cap \mathrm{SCOM}(g^{-1}, f^{-1}, g, f).$$

The pair (f, g) is termed *commutative* iff

$$\mathrm{COM}(f, g) = \Sigma^*.$$

We will address first the following problems (P1)–(P3). The problems were investigated in [3] in a more general set-up. We state the problems for *codes* and their left inverses.

(P1). Give examples showing that the commutativity and semi – commutativity sets may differ.

(P2). Give examples of f, g, f^{-1}, g^{-1}, such that

$$\mathrm{SCOM}(f^{-1}, g^{-1}, f, g) \neq \mathrm{SCOM}(g^{-1}, f^{-1}, g, f).$$

(P3). Give examples of f and g, and of their left inverses f_1^{-1}, f_2^{-1} and g_1^{-1}, g_2^{-1} such that the sets

$$\mathrm{SCOM}(f_1^{-1}, g_1^{-1}, f, g),\ \mathrm{SCOM}(f_1^{-1}, g_2^{-1}, f, g),$$

$$\mathrm{SCOM}(f_2^{-1}, g_1^{-1}, f, g),\ \mathrm{SCOM}(f_2^{-1}, g_2^{-1}, f, g)$$

are all different.

At a first glance, such examples might seem contra-intuitive.(This was at least partially due to the fact that the case of morphisms was treated very quickly in [3].) The point is that only the equation

$$f^{-1}f(w) = w,\ \text{for all } w \in \Sigma^*,$$

is required in the definition of f^{-1}: for words w outside the range of f, we are free to define f^{-1} as we please. In what follows words of the form $f(w)$ are referred to as *forced* as far as the definition of f^{-1} is concerned.

We will also consider some restrictions concerning the definition of f^{-1} for non-forced values. A very useful notion is that of a *smooth inverse*.

Assume that f is a *constant-length code*, that is, there is an integer $n_f \geq 1$ such that

$$|f(a)| = n_f, \text{ for all } a \in \Sigma.$$

For a constant-length code f, a left inverse f^{-1} is termed *smooth* iff f^{-1} is a (total) mapping of Σ^* into Σ^* satisfying the following conditions (i)–(iv).

(i) $f^{-1}f(a) = a$ for all $a \in \Sigma$. (This is of course obvious because f^{-1} is a left inverse. It is stated here just for the sake of completeness.)

(ii) $|f^{-1}(w)| = 1$ whenever $|w| = n_f$.

(iii) $f^{-1}(w) = \lambda$ whenever $|w| < n_f$.

(iv) $f^{-1}(w_1 \dots w_k u) = f^{-1}(w_1) \dots f^{-1}(w_k)$, whenever $k \geq 1$, $|w_1| = \dots = |w_k| = n_f$, $|u| < n_f$.

Theorem 1 *A smooth left inverse f^{-1} is uniquely determined by the values $f^{-1}(w)$, where $|w| = n_f$. It is a morphism exactly in case $n_f = 1$. In that case f^{-1} is also a code.*

Proof. The first sentence is clear by the definition because the decomposition of words longer than n_f in the form required in (iv) is unique. Assume that $n_f = 1$. Since f is a code, it permutes the letters of Σ. By (i) and (ii), f^{-1} must be the inverse permutation and hence, by (iii) and (iv), f^{-1} is a code. Assume, finally, that $n_f > 1$. Let w_1 and w_2 be words of lengths n_f and $n_f - 1$, respectively. Then

$$|f^{-1}(w_1 w_2)| = 1, \ f^{-1}(w_2) = \lambda, \ |f^{-1}(w_1 w_2 w_2)| = 2.$$

Consequently, $f^{-1}(w_1 w_2 w_2) \neq f^{-1}(w_1 w_2) f^{-1}(w_2)$, which implies that f^{-1} is not a morphism. □

The connection between semi-commutativity sets and equality sets is expressed by the equation

$$\mathrm{SCOM}(f^{-1}, g^{-1}, f, g) = E(f^{-1}g^{-1}fg, \ id),$$

valid in case $f^{-1}g^{-1}fg$ is a morphism. This requirement is satisfied for smooth left inverses.

Theorem 2 *Assume that f and g are constant length codes (possibly $n_f \neq n_g$), and that f^{-1} and g^{-1} are smooth. Then $f^{-1}g^{-1}fg$ is a letter-to-letter morphism.*

Proof. For $a \in \Sigma$, the lengths of $g(a)$, $fg(a)$, $g^{-1}fg(a)$ and $f^{-1}g^{-1}fg(a)$ are n_g, $n_g n_f$, n_f and 1, respectively. Thus, $f^{-1}g^{-1}fg$ maps letters into letters. To show that $f^{-1}g^{-1}fg$ is a morphism, it suffices to prove that, for $a \in \Sigma$ and $w \in \Sigma^*$, we have

$$(*) \qquad f^{-1}g^{-1}fg(aw) = f^{-1}g^{-1}fg(a) \ f^{-1}g^{-1}fg(w).$$

Because fg is a morphism, we obtain

$$f^{-1}g^{-1}fg(aw) = f^{-1}g^{-1}(fg(a)fg(w)).$$

Since always the first letter of $f^{-1}g^{-1}(x)$ is completely determined by the prefix of length $n_f n_g$ of x, we conclude that $f^{-1}g^{-1}fg(a)$ is the first letter on the right side and that, consequently, the claim $(*)$ follows. □

3 Illustrations of various constructions

In this section we will provide answers to the problems (P1)–(P3) by effectively constructing morphisms f, g and left inverses f^{-1}, g^{-1}, which satisfy the requested relations. The idea is namely to use the fact that, except for the forced values, we are free to define the left inverse of a code as we like. Consequently, we can define the non-forced values of the left inverses in a way that suits our purposes.

All the constructed morphisms will be constant-length codes and all the left inverses will be smooth.

We begin by considering (P1). The following theorem provides an example in which

$$\mathrm{SCOM}(f^{-1}, g^{-1}, f, g) \neq \mathrm{COM}(f, g).$$

Theorem 3 *There exist constant-length codes f, g, and smooth left inverses f^{-1}, g^{-1} such that (f^{-1}, g^{-1}, f, g) is semi-commutative, (f^{-1}, g^{-1}) is commutative, but (f, g) is not commutative.*

Proof. Let $\Sigma = \{a, b\}$ and the morphisms f, defined by:

$$f: \quad a \longrightarrow aa,\ b \longrightarrow bb,$$

$$g: \quad a \longrightarrow ab,\ b \longrightarrow ba.$$

The morphisms f, g are obviously constant-length codes with $n_f = n_g = 2$. Consider now the smooth left inverses f^{-1} and g^{-1} of f, respectively g:

$$\begin{array}{ll} f^{-1}: & aa \longrightarrow a,\ bb \longrightarrow b,\ ab \longrightarrow a,\ ba \longrightarrow b, \\ g^{-1} = & f^{-1}. \end{array}$$

Note that, according to Theorem 1, the listed values are enough to uniquely determine f^{-1}, g^{-1}.

The quadruple (f^{-1}, g^{-1}, f, g) is semi-commutative as for any word $w \in \Sigma^*$ we have:

$$f^{-1}g^{-1}fg(w) = g^{-1}f^{-1}fg(w) = g^{-1}g(w) = w.$$

(We have used the facts that $f^{-1} = g^{-1}$ and f^{-1} (resp. g^{-1}) is a left inverse of f (resp. g).)

The pair (f, g) is not commutative as we have:

$$fg(a) = a^2b^2 \neq abab = gf(a).$$

On the other hand, as $f^{-1} = g^{-1}$, the pair (f^{-1}, g^{-1}) is commutative. □

The following result shows that the sets $\mathrm{SCOM}(f^{-1}, g^{-1}, f, g)$ and $\mathrm{COM}(f^{-1}, g^{-1})$ may also differ. As in the preceding proof, the fact that the left inverses are smooth allows us to uniquely define them by listing only a finite number of values.

Theorem 4 *There exist constant-length codes f, g and smooth left inverses f^{-1}, g^{-1} such that (f^{-1}, g^{-1}, f, g) is semi-commutative, (f, g) is commutative but (f^{-1}, g^{-1}) is not commutative.*

Proof. Let $\Sigma = \{a, b\}$ and define the morphisms f, g by

$$f: \quad a \longrightarrow bab,\ b \longrightarrow aba,$$

$$g: \quad a \longrightarrow aba,\ b \longrightarrow bab.$$

It is easy to see that f, g are constant-length codes with $n_f = n_g = 3$.

Consider now the smooth left inverses f^{-1}, g^{-1} of f, respectively g,

$$\begin{array}{ll} f^{-1}: & bab \longrightarrow a,\ bw \longrightarrow b \text{ for } w \neq ab, |w| = 2, \\ & aba \longrightarrow b,\ aw \longrightarrow a \text{ for } w \neq ba, |w| = 2. \\ g^{-1}: & aw \longrightarrow a \text{ for } w \in \Sigma^2, \\ & bw \longrightarrow b \text{ for } w \in \Sigma^2. \end{array}$$

These values uniquely define the left inverses f^{-1}, g^{-1} (Theorem 1).

According to Theorem 2, in order to prove that $\mathrm{SCOM}(f^{-1}, g^{-1}, f, g) = \Sigma^*$ it suffices to show that the equality

$$f^{-1}g^{-1}fg(a) = a$$

holds for any letter $a \in \Sigma$.

In our case,

$$f^{-1}g^{-1}fg(a) = f^{-1}g^{-1}f(aba) = f^{-1}g^{-1}(bab\ aba\ bab) = f^{-1}(bab) = a,$$

$$f^{-1}g^{-1}fg(b) = f^{-1}g^{-1}f(bab) = f^{-1}g^{-1}(aba\ bab\ aba) = f^{-1}(aba) = b,$$

and therefore the (f^{-1}, g^{-1}, f, g) is semi-commutative.

The pair (f, g) is commutative as f, g are morphisms and the following equalities hold:

$$\begin{array}{llll} fg(a) = & f(aba) = & bab\ aba\ bab = & g(bab) = gf(a), \\ fg(b) = & f(bab) = & aba\ bab\ aba = & g(aba) = gf(b). \end{array}$$

On the other hand, (f^{-1}, g^{-1}) is not commutative as shown by the following example:

$$f^{-1}g^{-1}(aba\ aaa\ bbb) = f^{-1}(aab) = a \neq$$
$$g^{-1}f^{-1}(aba\ aaa\ bbb) = g^{-1}(bab) = b.$$

□

The problem (P1) is completely settled by the following theorem which provides morphisms f, g such that the set $\mathrm{SCOM}(f^{-1}, g^{-1}, f, g)$ differs from both $\mathrm{COM}(f, g)$ and $\mathrm{COM}(f^{-1}, g^{-1})$.

Note that, as in the preceding proof, the fact that f, g are constant-length codes and their left inverses are smooth helps in proving the semi-commutativity. Indeed, as in this case $f^{-1}g^{-1}fg$ is a morphism, it suffices to show that

$$f^{-1}g^{-1}fg(a) = a, \text{ for all } a \in \Sigma.$$

Theorem 5 *There exist constant-length codes f, g and smooth left inverses f^{-1}, g^{-1}, such that (f^{-1}, g^{-1}, f, g) is semi-commutative but neither (f, g) nor (f^{-1}, g^{-1}) is commutative.*

Proof. Let $\Sigma = \{a, b\}$ and define the morphisms f, g by:

$$f: \quad a \longrightarrow bbb,\ b \longrightarrow abb,$$

$$g: \quad a \longrightarrow aaa,\ b \longrightarrow aab.$$

Clearly, f, g are constant-length codes with $n_f = n_g = 3$.

Consider now the smooth left inverses f^{-1}, g^{-1},

$$\begin{aligned} f^{-1}: &\quad bbb \longrightarrow a,\ aab \longrightarrow b,\ bab \longrightarrow b,\ abb \longrightarrow b,\ bba \longrightarrow b,\\ &\quad w \longrightarrow a \text{ for any other } w \in \Sigma^3\\ g^{-1}: &\quad aaa \longrightarrow a,\ bbb \longrightarrow b,\ bab \longrightarrow a,\ aab \longrightarrow b,\ abb \longrightarrow a,\\ &\quad w \longrightarrow a \text{ for any other } w \in \Sigma^3. \end{aligned}$$

Theorem 1 assures that f^{-1}, g^{-1} are uniquely defined by the above listed values.

The quadruple (f^{-1}, g^{-1}, f, g) is semi-commutative. Indeed, for the letters of Σ we have

$$f^{-1}g^{-1}fg(a) = f^{-1}g^{-1}f(aaa) = f^{-1}g^{-1}(bbb\ bbb\ bbb) = f^{-1}(bbb) = a,$$

$$f^{-1}g^{-1}fg(b) = f^{-1}g^{-1}f(aab) = f^{-1}g^{-1}(bbb\ bbb\ abb) = f^{-1}(bba) = b.$$

According to Theorem 2, $f^{-1}g^{-1}fg$ is a letter-to-letter morphism and therefore the above equalities imply

$$f^{-1}g^{-1}fg(w) = w, \forall w \in \Sigma^*,$$

that is, (f^{-1}, g^{-1}, f, g) is semi-commutative.

On the other hand, (f, g) and (f^{-1}, g^{-1}) are not commutative as shown below:

$$fg(a) = f(aaa) = bbb\ bbb\ bbb \neq gf(a) = g(bbb) = aab\ aab\ aab,$$

$$f^{-1}g^{-1}(aab\ bbb\ bab) = f^{-1}(bba) = b \neq g^{-1}f^{-1}(aab\ bbb\ bab) = g^{-1}(bab) = a.$$

□

The order of the terms in $\mathrm{SCOM}(f^{-1}, g^{-1}, f, g)$ is quite essential. Indeed, the following theorem shows that changing the order affects the semi-commutativity set. This answers at the same time the problem (P2).

Theorem 6 *There exist constant-length codes f, g and smooth left inverses f^{-1}, g^{-1} such that (f^{-1}, g^{-1}, f, g) is semi-commutative but (g^{-1}, f^{-1}, g, f) is not.*

Proof. Consider the alphabet $\Sigma = \{a, b\}$ and the morphisms

$$f: \quad a \longrightarrow bbb,\ b \longrightarrow abb,$$

$$g: \quad a \longrightarrow aaa,\ b \longrightarrow aab.$$

It is clear that f, g are constant-length codes with $n_f = n_g = 3$.

Let us define now the smooth left inverses f^{-1}, g^{-1} of f respectively g by:

$$\begin{array}{ll} f^{-1}: & bbb \longrightarrow a,\ abb \longrightarrow b,\ aab \longrightarrow b,\ aab \longrightarrow b,\ bba \longrightarrow b, \\ & w \longrightarrow a, \text{ for other } w \in \Sigma^3, \\ g^{-1}: & aaa \longrightarrow a,\ aab \longrightarrow b,\ bbb \longrightarrow b,\ abb \longrightarrow a, \\ & w \longrightarrow a \text{ for other } w \in \Sigma^3. \end{array}$$

For the letters from Σ we have

$$f^{-1}g^{-1}fg(a) = f^{-1}g^{-1}f(aaa) = f^{-1}g^{-1}(bbb\ bbb\ bbb) = f^{-1}(bbb) = a,$$

$$f^{-1}g^{-1}fg(b) = f^{-1}g^{-1}f(aab) = f^{-1}g^{-1}(bbb\ bbb\ abb) = f^{-1}(bba) = b.$$

Consequently, using the fact that $f^{-1}g^{-1}fg$ is a morphism (Theorem 2) we deduce that $\mathrm{SCOM}(f^{-1}, g^{-1}, f, g) = \Sigma^*$.

On the other hand, $\mathrm{SCOM}(g^{-1}, f^{-1}, g, f) \neq \Sigma^*$ as a does not belong to $\mathrm{SCOM}(g^{-1}, f^{-1}, g, f)$:

$$g^{-1}f^{-1}gf(a) = g^{-1}f^{-1}g(bbb) = g^{-1}f^{-1}(aab\ aab\ aab) = g^{-1}(bbb) = b \neq a.$$

□

We will conclude this section with a somewhat unexpected result. For a fixed choice of morphisms f, g one can choose left inverses f_1^{-1}, f_2^{-1}, respectively g_1^{-1}, g_2^{-1} such that $(f_1^{-1}, g_1^{-1}, f, g)$ is semi-commutative, but all the semi-commutativity sets $\mathrm{SCOM}(f_i^{-1}, g_j^{-1}, f, g)$, $i, j \in \{1, 2\}$ are paiwise distinct.

Theorem 7 *There exist constant-length codes f, g and their smooth left inverses $f_1^{-1}, f_2^{-1}, g_1^{-1}, g_2^{-1}$ such that the sets*

$$SCOM(f_1^{-1}, g_1^{-1}, f, g), SCOM(f_1^{-1}, g_2^{-1}, f, g),$$

$$SCOM(f_2^{-1}, g_1^{-1}, f, g), SCOM(f_2^{-1}, g_2^{-1}, f, g),$$

are all different.

Proof. Let $\Sigma = \{a, b\}$ and f, g be the morphisms defined by

$$f: \ a \longrightarrow bab,\ b \longrightarrow aba,$$

$$g: \ a \longrightarrow aaa,\ b \longrightarrow bbb.$$

It is easy to see that f, g are constant-length codes with $n_f = n_g = 3$.

Consider now the following smooth left inverses of f:

$$\begin{array}{ll} f_1^{-1}: & bab \longrightarrow a,\ aba \longrightarrow b,\ aaa \longrightarrow a,\ bbb \longrightarrow b, \\ & w \longrightarrow a \text{ for any other } w \in \Sigma^3, \\ f_2^{-1}: & bab \longrightarrow a,\ aba \longrightarrow b,\ aaa \longrightarrow b,\ bbb \longrightarrow a, \\ & w \longrightarrow a \text{ for any other } w \in \Sigma^3, \end{array}$$

and the smooth left inverses of g:

$$\begin{array}{ll} g_1^{-1}: & aaa \longrightarrow a,\ bbb \longrightarrow b,\ bab \longrightarrow a,\ aba \longrightarrow b, \\ & w \longrightarrow a \text{ for any other } w \in \Sigma^3, \\ g_2^{-1}: & aaa \longrightarrow a,\ bbb \longrightarrow b,\ bab \longrightarrow b,\ aba \longrightarrow b, \\ & w \longrightarrow a \text{ for any other } w \in \Sigma^3. \end{array}$$

In order to prove the theorem, we will make use of the following facts (i) –(iv).

(i) $\mathrm{SCOM}(f_1^{-1}, g_1^{-1}, f, g) = \Sigma^*$. Indeed, we have

$$f_1^{-1} g_1^{-1} fg(a) = f_1^{-1} g_1^{-1} f(aaa) = f_1^{-1} g_1^{-1} (bab\ bab\ bab) = f_1^{-1}(aaa) = a,$$

and

$$f_1^{-1} g_1^{-1} fg(b) = f_1^{-1} g_1^{-1} f(bbb) = f_1^{-1} g_1^{-1} (aba\ aba\ aba) = f_1^{-1}(bbb) = b.$$

These equalities, together with Theorem 2 prove the requested relation.

(ii) $a, b \notin \mathrm{SCOM}(f_2^{-1}, g_1^{-1}, f, g)$. This relation is proved by the following equalities:

$$f_2^{-1} g_1^{-1} fg(a) = f_2^{-1}(aaa) = b,$$

$$f_2^{-1}g_1^{-1}fg(b) = f_2^{-1}(bbb) = a.$$

(iii) $b \in \mathrm{SCOM}(f_1^{-1}, g_2^{-1}, f, g)$ but $a \notin \mathrm{SCOM}(f_1^{-1}, g_2^{-1}, f, g)$. The relations hold true as we have

$$f_1^{-1}g_2^{-1}fg(b) = f_1^{-1}g_2^{-1}(aba\ aba\ aba) = f_1^{-1}(bbb) = b.$$

$$f_1^{-1}g_2^{-1}fg(a) = f_1^{-1}g_2^{-1}(bab\ bab\ bab) = f_1^{-1}(bbb) = b,$$

(iv) $a \in \mathrm{SCOM}(f_2^{-1}, g_2^{-1}, f, g)$ but $b \notin \mathrm{SCOM}(f_2^{-1}, g_2^{-1}, f, g)$. This is proved by the following equalities:

$$f_2^{-1}g_2^{-1}fg(a) = f_2^{-1}g_2^{-1}(bab\ bab\ bab) = f_2^{-1}(bbb) = a,$$

$$f_2^{-1}g_2^{-1}fg(b) = f_2^{-1}g_2^{-1}(aba\ aba\ aba) = f_2^{-1}(bbb) = a.$$

Using the relations (i) –(iv) we can deduce:
(i) + (ii) $\Longrightarrow \mathrm{SCOM}(f_1^{-1}, g_1^{-1}, f, g) \neq \mathrm{SCOM}(f_2^{-1}, g_1^{-1}, f, g)$
(i) + (iii) $\Longrightarrow \mathrm{SCOM}(f_1^{-1}, g_1^{-1}, f, g) \neq \mathrm{SCOM}(f_1^{-1}, g_2^{-1}, f, g)$
(i) + (iv) $\Longrightarrow \mathrm{SCOM}(f_1^{-1}, g_1^{-1}, f, g) \neq \mathrm{SCOM}(f_2^{-1}, g_2^{-1}, f, g)$
(ii) + (iii) $\Longrightarrow \mathrm{SCOM}(f_2^{-1}, g_1^{-1}, f, g) \neq \mathrm{SCOM}(f_1^{-1}, g_2^{-1}, f, g)$
(ii) + (iv) $\Longrightarrow \mathrm{SCOM}(f_2^{-1}, g_1^{-1}, f, g) \neq \mathrm{SCOM}(f_2^{-1}, g_2^{-1}, f, g)$
(iii) + (iv) $\Longrightarrow \mathrm{SCOM}(f_1^{-1}, g_2^{-1}, f, g) \neq \mathrm{SCOM}(f_2^{-1}, g_2^{-1}, f, g)$
and the proof of the theorem is complete. □

4 Smooth left inverses

Recall that in Theorem 2 we proved that if f, g are constant-length codes and $f^{-1}g^{-1}$ are smooth then $f^{-1}g^{-1}fg$ is a letter-to-letter morphism.

The following corollary of Theorem 2 gives an exhaustive characterization of semi-commutativity sets in case of smooth left inverses.

Theorem 8 *Assume that f, g, f^{-1}, g^{-1} are as in Theorem 2. Then*

$$SCOM(f^{-1}, g^{-1}, f, g) = \Sigma_1^*, \text{ for some } \Sigma_1 \subseteq \Sigma.$$

(For $\Sigma_1 = \emptyset$, it is agreed that $\Sigma_1^* = \{\lambda\}$.)

It is important to observe that, even in the set-up of Theorems 2 and 8, $f^{-1}g^{-1}fg$ is not necessarily injective. In other words, although it is a morphism, it is not necessarily a code. We consider the following example.

$$f: \ a \longrightarrow aba,\ b \longrightarrow abb \qquad (n_f = 3),$$

$$g: \ a \longrightarrow aa,\ b \longrightarrow bb \qquad (n_g = 2).$$

As regards arguments of lengths n_f and n_g, the forced values of f^{-1} and g^{-1} are:

$$f^{-1}(aba) = a,\ f^{-1}(abb) = b,\ g^{-1}(aa) = a,\ g^{-1}(bb) = b.$$

Hence, we are free to define

$$g^{-1}(ab) = b,\ g^{-1}(ba) = a.$$

Consequently,

$$g^{-1}fg(a) = g^{-1}(abaaba) = baa,$$
$$g^{-1}fg(b) = g^{-1}(abbabb) = bab.$$

We are also free to define

$$f^{-1}(baa) = f^{-1}(bab) = a,$$

yielding

$$f^{-1}g^{-1}fg(a) = f^{-1}g^{-1}fg(b) = a,$$

which shows that $f^{-1}g^{-1}fg$ is not injective.

It is obvious that the construction of this example can be generalized to yield the following result.

Theorem 9 *For any letter-to-letter morphism $h : \Sigma^* \longrightarrow \Sigma^*$, constant-length codes f and g, as well as their smooth left inverses f^{-1} and g^{-1}, with the property*

$$f^{-1}g^{-1}fg = h$$

can be effectively constructed.

5 Considerations related to smoothness

No necessary and sufficient conditions are known for $f^{-1}g^{-1}fg$ to be a morphism. In particular, Theorem 2 does not yield a necessary and sufficient condition. We now investigate matters related to this issue. We begin with a result concerning the question about a left inverse being a morphism.

Theorem 10 *Whenever a left inverse h^{-1} of a morphism $h : \Sigma^* \longrightarrow \Sigma^*$ is itself also a morphism, then it permutes the letters of Σ and, hence, is a code.*

Proof. Since h has a left inverse, it must be a code and, consequently, nonerasing. Hence, if $\Sigma = \{a_1, \ldots, a_n\}$ we have for all i

$$h(a_i) = a_{j_1} \ldots a_{j_{k(i)}},\ \ k(i) \geq 1.$$

Because h^{-1} is a left inverse, we have

$$h^{-1}h(a_i) = a_i,\ \ 1 \leq i \leq n.$$

Because h^{-1} is a morphism, we have

$$h^{-1}h(a_i) = h^{-1}(a_{j_1} \dots a_{j_{k(i)}}) = h^{-1}(a_{j_1}) \dots h^{-1}(a_{j_{k(i)}}).$$

Consequently, for all i,

$$a_i = h^{-1}(a_{j_1}) \dots h^{-1}(a_{j_{k(i)}}).$$

This is possible only if a_i equals some of the factors on the right side:

$$a_i = h^{-1}(a_{\alpha(i)}), \;\; 1 \leq \alpha(i) \leq n.$$

Here α is a permutation of the set $\{1, \dots, n\}$, because if $\alpha(i) = \alpha(j)$ for $i \neq j$, then h^{-1} would map $a_{\alpha(i)}$ to both a_i and a_j, which is impossible. This implies that h^{-1} does not map any letters to λ and, consequently, $k(i) = 1$. We have shown that h^{-1} is letter-to-letter and a code. □

We show next that the converse of Theorem 2 does not hold true. Indeed, $f^{-1}g^{-1}fg$ may be a letter-to-letter morphism (even the identity morphism), although neither f nor g is constant-length. We need the following auxiliary result in our example.

Lemma 1 *Every word $w \in \{a, b\}^*$ posseses a unique decomposition*

$$w = w_1 \dots w_k w_{k+1}, \;\; k \geq 0,$$

where $w_i \in \{b, aa, ab\}$, $1 \leq i \leq k$, and $w_{k+1} \in \{a, \lambda\}$.

Proof. We read an arbitrary given w from left to right. If w begins with b, we get a unique left factor $w_i = b$, and may proceed with the shorter word. If $w = a$ or $w = \lambda$, we get a unique w_{k+1} ($k = 0$ if this alternative holds initially), and are through. Otherwise, w must begin either with aa or ab. In both cases we get a unique left factor w_i and may proceed with a shorter word. □

Consider now the morphism g defined by

$$g: \;\; a \longrightarrow aa, \;\; b \longrightarrow b.$$

The values

$$g^{-1}(aa) = a, \;\; g^{-1}(b) = b$$

are forced. Moreover, $g^{-1}(x)$ is forced for all $x \in \{aa, b\}^*$, and is obtained by the appropriate catenation of a and b. (Observe that also $g^{-1}(\lambda) = \lambda$ is forced; this statement holds for all morphisms.)

We are free to define the values

$$g^{-1}(ab) = b, \;\; g^{-1}(a) = \lambda.$$

We now write an arbitrary $w \in \{a,b\}^*$ as in Lemma 1, and define

$$(*) \qquad g^{-1}(w) = g^{-1}(w_1 \dots w_k w_{k+1}) = g^{-1}(w_1) \dots g^{-1}(w_k) g^{-1}(w_{k+1}).$$

Thus, g^{-1} is total. For forced argument values, this definition coincides with the one given above. Observe also that always $g^{-1}(w_{k+1}) = \lambda$.

The morphism f is defined in the same way, with a and b interchanged. Explicitly,

$$f: \ a \longrightarrow a, \ b \longrightarrow bb,$$

$$f^{-1}: \ a \longrightarrow a, \ bb \longrightarrow b, \ ba \longrightarrow a, \ b \longrightarrow \lambda.$$

The definition $(*)$ holds for f^{-1} as well, with the corresponding change in Lemma 1: the two sets of the lemma are now $\{a, bb, ba\}$ and $\{b, \lambda\}$.

Consider the value $f^{-1}g^{-1}fg(w)$. First fg doubles every letter in w. Then g^{-1} removes the doubling from the a's but keeps it for the b's until, finally, f^{-1} removes it also from the b's. Consequently, $f^{-1}g^{-1}fg$ is the identity morphism. However, neither f nor g is constant length.

In this example, the non-forced values were not actually needed. However, the example serves also as an introduction to the following definition. We first generalize the situation of Lemma 1.

Let Σ be an alphabet, and M_1, M_2 two finite subsets of Σ^*. The pair (M_1, M_2) is termed a *residual code* for Σ iff every word $w \in \Sigma^*$ possesses a unique decomposition

$$w = w_1 \dots w_k w_{k+1}, \ \ k \geq 0,$$

where $w_i \in M_1$, $1 \leq i \leq k$, and $w_{k+1} \in M_2$.

The following results are immediate from the definition.

Lemma 2 *If (M_1, M_2) is a residual code for Σ, then M_1 is a code, $\lambda \notin M_1$ and $\lambda \in M_2$.*

The proof of the next lemma is analogous to the proof of Lemma 1.

Lemma 3 *The pair (M_1, M_2) (resp. (M_1', M_2')), where*

$$M_1 = \{baa, bba, bbb, bab, aa, ab\}, M_2 = \{ba, bb, a, b, \lambda\}$$

$$\textit{(resp. } M_1' = \{aaa, aab, aba, abb, ba, bb\}, M_2' = \{aa, ab, a, b, \lambda\}\textit{)}$$

is a residual code for $\{a, b\}$.

We are now ready for the definition of semi-smoothness. A left inverse f^{-1} of a morphism $f: \Sigma^* \longrightarrow \Sigma^*$ is termed *semi-smooth* iff there is a residual code (M_1, M_2) for Σ such that the following conditions (i)–(iv) are satisfied.

(i) For all $a \in \Sigma$, $f(a) \in M_1$.

(ii) For all $w \in M_1$, $|f^{-1}(w)| = 1$.

(iii) For all $w \in M_2$, $f^{-1}(w) = \lambda$.

(iv) $f^{-1}(w_1 \dots w_k w_{k+1}) = f^{-1}(w_1) \dots f^{-1}(w_k)$

whenever $k \geq 1$, $w_i \in M_1$, for $1 \leq i \leq k$ and $w_{k+1} \in M_2$.

Observe that (i)–(iv) above correspond to (i)–(iv) in the definition of smoothness. Because f has a left inverse, it is a code and, therefore, the words $f(a)$ in (i) are distinct. If we let M_1 to consist of all words of length n, for some $n \geq 1$, and M_2 to consist of all words shorter than n, then (M_1, M_2) is a residual code for Σ. Hence, we obtain the following result.

Theorem 11 *Every smooth left inverse is semi-smooth but not vice versa.*

The left inverses g^{-1} and f^{-1} defined after Lemma 1 are semi-smooth. They cannot be smooth because neither g nor f is constant-length.

We now return to Theorem 2. It turns out that the essential requirement is that the words in M_1 are of equal length. The following theorem shows that not even semi-smoothness together with the morphisms being constant-length suffices.

Theorem 12 *There are constant-length codes f and g with semi-smooth left inverses f^{-1} and g^{-1} such that $f^{-1}g^{-1}fg$ is not a morphism.*

Proof. Choose $\Sigma = \{a, b\}$. Define first f and g by

$$g: \quad a \longrightarrow baa, \; b \longrightarrow bbb,$$

$$f: \quad a \longrightarrow aa, \; b \longrightarrow abb.$$

We now apply Lemma 3, and define semi-smooth inverses g^{-1} and f^{-1}, using the residual code (M_1, M_2) in connection with g^{-1}, and (M_1', M_2') in connection with f^{-1}. Since, for argument values in M_2 and M_2', the values of the inverses equal λ, it suffices to define the inverses for argument values in M_1 and M_1':

$$g^{-1}: \quad baa \longrightarrow a, \; bbb \longrightarrow b, \; bba \longrightarrow a, \; bab \longrightarrow a, \; aa \longrightarrow b, \; ab \longrightarrow b,$$

$$f^{-1}: \quad aaa \longrightarrow a, \; abb \longrightarrow b, \; aab \longrightarrow a, \; aba \longrightarrow b, \; bb \longrightarrow a, \; ba \longrightarrow b.$$

(Of course, (iv) in the definition of the semi-smoothness takes care of the remaining argument values.)

It is now immediately verified that

$$f^{-1}g^{-1}fg(bb) = ba \neq bb = f^{-1}g^{-1}fg(b)f^{-1}g^{-1}fg(b).$$

Hence, $f^{-1}g^{-1}fg$ is not a morphism. □

6 Unrestricted left inverses

We use the general term *unrestricted* for left inverses of codes $h : \Sigma^* \longrightarrow \Sigma^*$ if nothing is assumed concerning how the non-forced values are defined. It is not even required that the left inverse is a total function. Of course, we can always make it total by defining, for instance, $h^{-1}(w) = \lambda$ for otherwise undefined argument values w. Such a definitional extension will not cause any difficulties because no morphic structure is assumed.

Rather bizarre constructions become possible for unrestricted left inverses. In some sense the following result represents the culmination of such constructions.

Theorem 13 *Given an alphabet $\Sigma = \{a_1, \ldots, a_n\}$, $n \geq 2$, there are codes $f, g : \Sigma^* \longrightarrow \Sigma^*$ with the following property. For every language $L \subseteq \Sigma^*$ (posibly not even recursively enumerable), a left inverse ${g_L}^{-1}$ of g may be defined such that, for every left inverse f^{-1} of f,*

$$SCOM(f^{-1}, g^{-1}, f, g) = L \cup \{\lambda\}.$$

Proof. The codes g and f are defined by

$$g: \quad a_i \longrightarrow a_2 {a_1}^i, \ 1 \leq i \leq n,$$

$$f: \quad a_i \longrightarrow {a_1}^i a_2, \ 1 \leq i \leq n.$$

Clearly, only λ from the range of g is also in the range of f. Hence, when dealing with unrestricted left inverses g^{-1}, we may choose the value $g^{-1}(w)$ freely for $w = f(u)$, $u \neq \lambda$.

We now define

$${g_L}^{-1}(f(g(w))) = f(w) \text{ for } w \in L.$$

(Observe that this equation holds also for $w = \lambda$.) For other argument values (except the forced ones), ${g_L}^{-1}$ is undefined. It is immediate that the theorem holds for ${g_L}^{-1}$ thus defined. □

Observe that g and f are fixed, and that only forced values of f^{-1} are needed. All the complexities of L are embedded in ${g_L}^{-1}$.

7 Undecidability

For decision problems it is necessary to assume that the left inverses are effectively given. This is certainly true as regards ${g_L}^{-1}$ if the language L is recursive because then, for any w, we may effectively compute the value $g^{-1}(w)$. The following summarizes some of the undecidability results obtainable because of Theorem 13.

Theorem 14 *Each of the following problems is undecidable for an effectively given semi-commutativity set*

$$S = SCOM(f^{-1}, g^{-1}, f, g).$$

Is S nonempty? (This means that $S \neq \{\lambda\}$.) Is S finite? Is S regular? Is S context-free? Is $S = \Sigma^*$? Is $S = K$ where K is some fixed language?

Proof. Let L be given by a context-sensitive grammar in Theorem 13. Then L is recursive. The theorem follows because all of the problems mentioned before are undecidable for context-sensitive grammars. □

8 Conclusions

The original motivations for considering semi- commutativity sets come from cryptography. We believe that we have shown the importance of these sets in language theory. We have settled the basic problems concerning semi- commutativity sets of morphisms, and investigated the sets both in case of restricted and unrestricted left inverses. In our estimation, an important open research area is the study of left inverses and the corresponding semi-commutativity sets under the restriction that only words of length ≤ 1 can be used as "basic" values $f^{-1}(w)$. This is not the case in the definition of Theorem 13.

References

[1] G.Brassard, C.Crepeau and J.-M. Robert. All-or-Nothing Disclosure of Secrets. *Springer Lecture Notes in Computer Science*, 263 (1987) 234-238.

[2] H.Nurmi, A.Salomaa and L.Santean. Secret Ballot Elections in Computer Networks. *Computers & Security*, 10 (1991) 553-560.

[3] G.Păun and A.Salomaa. Semi-commutativity sets – a cryptographically grounded topic. *Bull.Math. Soc.Sci.Math Roumaine*.

[4] A.Renvall. ANDOS: a simple protocol for secret selling of secrets. *EATCS Bulletin*, 47 (1992).

[5] A.Salomaa. *Jewels of Formal Language Theory*. Computer Science Press, Rockville, Maryland (1981).

[6] A.Salomaa. *Public-Key Cryptography*. Springer-Verlag, Berlin, Heidelberg, New York (1990).

[7] A.Salomaa and L.Santean. Secret Selling of Secrets with Several Buyers. *EATCS Bulletin*, 42 (1990), 178-186.