Theoretical Computer Science 116 (1993) 339–357
Elsevier

Language-theoretic problems arising from Richelieu cryptosystems

Mircea Andrașiu, Gheorghe Păun
Institute of Mathematics, Str. Academiei 14, București 70109, Romania

Jürgen Dassow
Magdeburg University of Technology, Department of Mathematics PO BOX 124, O-3010 Magdeburg, Germany

Arto Salomaa
Academy of Finland and Mathematics Department, University of Turku, 20500 Turku, Finland

Communicated by M. Nivat
Received August 1991
Revised December 1991

Abstract

Andrașiu, M., J. Dassow, Gh. Păun and A. Salomaa, Language-theoretic problems arising from Richelieu cryptosystems, Theoretical Computer Science 116 (1993) 339–357.

Classical cryptosystems of the type "Richelieu" or "garbage-in-between" give rise to ideas and notions so far neglected in the theory of formal languages. This paper investigates such notions, mainly from the point of view of decidability.

1. Introduction and basic notions

The theory of formal languages provides a suitable framework for discussing cryptosystems. Indeed, the operations defined by cryptosystems can be viewed as mappings between sets of words. Thus, the study of such operations is a part of language theory. In many cases, where the cryptographic mappings are based on number theory, the methods of traditional language theory are not applicable. On

Correspondence to: A. Salomaa, Department of Mathematics, University of Turku, 20500 Turku, Finland.

0304-3975/93/$06.00

the other hand, language-theoretic notions can be used as a basis for public-key cryptosystems [4]. Further study of such notions is called for because cryptography should not become dangerously dependent on the computational complexity of the few number-theoretic problems. For instance, [7] is a step in this direction.

In the cryptosystem customarily attributed to *Richelieu* (and used also in the novel "Mathias Sandorf" by Jules Verne), both the sender and the legal receiver have identical sheets of cardboard with holes. When the sheet is positioned on top of the ciphertext, the plaintext becomes visible through the holes. This is a special case of the cryptographic method referred to as *garbage-in-between* [4]. Only some letters of the ciphertext, specified according to their position or by some other means, are significant, the remaining letters being just garbage. In the past this method was frequently used to make the ciphertext look something innocent.

Embedding a word x into a word y, which is exactly the cryptographic idea of garbage-in-between, corresponds to the language-theoretic *shuffle operation* [1, 2]. Let x and y be words over an alphabet V. By definition,

$$\begin{aligned}\mathrm{Shuf}(x, y) = \{ & x_1 y_1 x_2 y_2 \dots x_n y_n \mid \\ & x = x_1 x_2 \dots x_n,\ y = y_1 y_2 \dots y_n,\ n \geqslant 1, \\ & \text{all } x_i \text{ and } y_i \text{ in } V^* \}.\end{aligned}$$

We can visualize x as the plaintext, that is, the word visible through the holes in the cardboard sheet. Shuffle operation applied to languages, $\mathrm{Shuf}(L_1, L_2)$, will also be considered in the sequel. For undefined notions in language theory, [5] may be consulted.

The decryption according to the Richelieu cryptosystem (the role of the holes in the sheet) can be modelled by the operation of *guided filtering*. Let x and y be words of equal length over the alphabets V and $\{0, 1\}$, respectively. Denote by $|x|$, $s_i(x)$ and $N_a(x)$ the length of x, the ith letter of x, $1 \leqslant i \leqslant |x|$, and the number of occurrences of the letter a in x, respectively. If $|x| = |y| = n$, then the *guided filtering of x through y* is defined by $\mathrm{GF}(x, y) = a_1 a_2 \dots a_n$, where, for $1 \leqslant i \leqslant n$,

$$a_i = \begin{cases} s_i(x) & \text{if } s_i(y) = 1, \\ \lambda & \text{if } s_i(y) = 0. \end{cases}$$

Intuitively, x is the ciphertext and y is the sheet, where the holes are in the positions indicated by occurrences of the letter 1. If y consists of 0's, there are no holes, and the result of the decryption is the empty word λ.

The operation of guided filtering, similar to the operation of parallel controlled deletion considered in [7], is extended to languages in the natural way:

$$\mathrm{GF}(L_1, L_2) = \{\mathrm{GF}(x, y) \mid x \in L_1,\ y \in L_2,\ |x| = |y|\},$$

where $L_1 \subseteq V^*$, $L_2 \subseteq \{0, 1\}^*$.

The word y (or several words y) can be viewed as the decryption key: plaintext results when guided filtering is applied to the ciphertext and key. Such an application

can obviously be carried out in real time. As usual in classical cryptography, in addition to security considerations, one has to deal here with problems of key management. Is it possible to recover the plaintext from the middle of garbage without using a filtering key? This, of course, depends on the encryption method. It is desirable that the effect of shuffling be reversed even without the exact knowledge of a key.

Assume that the messages x are encrypted by scrambling them with the word y, the result of the scramble being the ciphertext z. Thus, a desirable situation would be that x is (fairly easily) recovered from z without the exact knowledge of y. Cryptosystems "transparent" in this sense are classical rather than public-key because the exact encryption method cannot be publicized – an eavesdropper should not be able to recover x from z.

We now describe such a transparent cryptosystem. The study of the resulting language-theoretic notions has so far been neglected, although the notions are rather interesting.

Consider words over the alphabet V, and let U be a subalphabet of V. Extend the notation $N_a(x)$ to concern subalphabets by

$$N_U(x)=\sum_{a\in U} N_a(x).$$

The operation of *guided sparse substitution* is defined for words x and y satisfying $N_U(y)\geqslant|x|=k$ as follows:

$$\mathrm{GSS}(y, x)=y_1a_1y_2a_2\ldots y_ka_ky_{k+1}$$

for $k\geqslant 1$, $y_i\in(V-U)^*$, $1\leqslant i\leqslant k$, $y_{k+1}\in V^*$, provided

$$y=y_1b_1y_2b_2\ldots y_kb_ky_{k+1}, \quad b_i\in U,$$

$$x=a_1a_2\ldots a_k, \quad a_i\in V \text{ for all } i.$$

Assume now that the words y used as encryption keys come from a language L over the alphabet V such that L contains at most one word of any given length n, possibly there being finitely many exceptional values of n. Assume further that the legal receiver knows L and the designated subset U. Given a ciphertext $\mathrm{GSS}(y, x)=z$, the legal receiver is able to find y since $|y|=|z|$. If $|z|$ is one of the exceptional values of n, the legal receiver has to try finitely many possible y's. Of course, the exceptional values of n may also be avoided in the encryption.

The occurrences of the letters of U in y indicate the "position of the holes in the cardboard". The plaintext can be read from the holes. Since we have only $N_U(y)\geqslant|x|$ and not necessarily $N_U(y)=|x|$, we obtain in this fashion a word xw, where the original plaintext occurs as a prefix. Which prefix is actually intended has to be found out in some other fashion, such as considering meaningfulness. From the point of view of secrecy, it would be too dangerous to require $N_U(y)=|x|$.

We are now ready for the language-theoretic definitions basic for this paper.

A language L over the alphabet V is called *thin* if, for almost all n,

$$\text{card}(L \cap V^n) \leqslant 1.$$

A subset L_1 of a language L is termed *length-complete* for L if, whenever L contains a word of length n, then also L_1 contains a word of length n. If, moreover, no proper subset of L_1 is length-complete for L, L_1 is termed *minimal length-complete*, abbreviated MLC for L.

Thus, every language is length-complete for itself. An MLC language cannot contain two words of the same length and, consequently, is thin. Even if a language L is not thin, MLC subsets of L can be used for cryptographic purposes in the way described above.

This paper investigates problems arising in a natural way from the notions defined above. The technical contributions will be language-theoretic rather than cryptographic. We just want to point out interconnections with cryptography and hope to return in another context to the cryptographic issues involved.

2. Preliminary results

The following are typical examples of thin languages:

$$L_1 = (abc)^*,$$

$$L_2 = \{a^n b^n \mid n \geqslant 0\},$$

$$L_3 = \{a^n b^n c^n \mid n \geqslant 0\},$$

$$L_4 = \{a^p \mid p \text{ is a prime number}\},$$

$$L_5 = \{x \in \{0, 1\}^* \mid x \text{ is a prefix in the binary representation of } \pi\}.$$

Here L_1 is regular, L_2 linear but not regular, L_3 and L_4 context-sensitive but not context-free (see [5]). The latter statement is most likely to hold also for L_5, although we do not know of any documentation. Each of the languages L_1–L_5 is MLC for itself. We do not know any nonlinear context-free thin languages. Indeed, we conjecture that there are no such languages

Fundamental problems concerning thinness are the following. Consider languages belonging to a specific language class, such as a class in the Chomsky hierarchy. Is it decidable whether or not a given language is thin? Does every language possess a MLC subset in the same class? Is the construction of such a subset effective? These problems will be attacked in Sections 3–5.

Observe that simple languages may possess MLC subsets of arbitrarily high complexity. For instance, if T is any subset of the set $\mathbf{N}$ of nonnegative integers, then the language

$$\{a^n \mid n \in T\} \cup \{b^n \mid n \in \mathbf{N} - T\}$$

is a MLC subset of $a^* \cup b^*$. The same idea can be used to show that any nonthin language possesses MLC subsets of arbitrarily high complexity.

There are also many other natural problems concerning thinness in addition to the fundamental problems listed above. For instance, from the point of view of cryptography, it is desirable to construct easy-to-handle classes of thin languages. However, in this paper attention will be restricted to the problems listed above.

We will now establish some basic facts concerning the operations GF and GSS. The alphabets will be as in the definitions.

Lemma 2.1. *Guided filtering can be expressed as*

$$\mathrm{GF}(L_1, L_2) = g(\mathrm{Shuf}(L_1, L_2) \cap (\{0, 1\} V)^*),$$

where g is a gsm mapping.

Proof. The gsm

$$g = (\{s_0, s_1\}, V \cup \{0, 1\}, V, s_0, P, \{s_0\}),$$

$$P = \{s_0 0 \to s_0, s_0 1 \to s_1\}$$

$$\cup \{s_0 a \to s_0 \mid a \in V\}$$

$$\cup \{s_1 a \to a s_0 \mid a \in V\}$$

satisfies the equation. □

The following result is now obvious by known closure properties.

Theorem 2.2. *The families of regular and of recursively enumerable languages are closed under GF; the families of linear and of context-free languages are closed under guided filtering through regular languages.*

Theorem 2.3. (i) *The families of linear and of context-free languages are not closed under GF operation;* (ii) *the family of context-sensitive languages is not closed under GF through regular sets.*

Proof. (i) Take the linear languages

$$L_1 = \{a^n c b^n c a^m \mid n, m \geqslant 1\},$$

$$L_2 = \{1^m 0 1^n 0 1^n \mid n, m \geqslant 1\}.$$

We clearly have

$$\mathrm{GF}(L_1, L_2) \cap a^* b^* a^* = \{a^n b^n a^n \mid n \geqslant 1\},$$

which is not a context-free language; hence, also $\mathrm{GF}(L_1, L_2)$ is not context-free.

(ii) For each recursively enumerable language $L \subseteq V^*$, there is a context-sensitive language $L' \subseteq V^*\{b\}a^*$, a, b symbols not in V, such that $x \in L$ iff $xba^n \in L'$ for some $n \geqslant 0$. Therefore,

$$\mathrm{GF}(L', 1^*0^*) \cap V^*\{b\} = L\{b\};$$

hence, for noncontext-sensitive L, $\mathrm{GF}(L', 1^*0^*)$ is noncontext-sensitive too, although L' is context-sensitive and 1^*0^* is regular. □

Analogous results can be established for guided sparse substitution.

Lemma 2.4. *Guided sparse substitution can be expressed as*

$$\mathrm{GSS}(L_1, L_2) = g(\mathrm{Shuf}(L_1, h(L_2)) \cap R),$$

where g is a gsm mapping, h is a morphism and R is a regular language.

Proof. Take

$$V' = \{a' \mid a \in V\},$$

$$h: V^* \to V'^*, \quad h(a) = a', \ a \in V,$$

$$R = ((V - U)^* U V')^* V^*,$$

$$g = (\{s_0, s_1, s_2\}, V \cup V', V, s_0, P, \{s_0, s_2\}),$$

with

$$P = \{s_0 a \to a s_0 \mid a \in V - U\}$$
$$\cup \{s_0 a \to s_1 \mid a \in U\}$$
$$\cup \{s_1 b' \to b s_0 \mid b \in V\}$$
$$\cup \{s_0 a \to a s_2 \mid a \in V\}$$
$$\cup \{s_2 a \to a s_2 \mid a \in V\}.$$

The morphism h marks the symbols, Shuf mixes the symbols, the intersection with R selects only the strings whose symbols in V' appear in pairs with symbols in U and, finally, g erases all symbols, in U followed by symbols in V' and replaces a' by a, $a \in V$. In conclusion, we have the equality in the statement. □

Theorem 2.5. *The families of regular, context-sensitive and recursively enumerable languages are closed under the GSS operation; the family of context-free languages is closed under GSS with regular languages.*

Proof. Follows directly from the closure properties of these families, observing that the morphism h is λ-free and the gsm g erases a linearly bounded number of symbols (this is important for the context-sensitive case). □

Lemma 2.6. *For all* $L_1, L_2, L_i \subseteq V^*$, $i=1, 2$, *we have*

$$L_1 \cap L_2 = h_1(GSS(h_2(L_1), h_3(L_2)) \cap R),$$

where h_1, h_2, h_3 *are morphisms and* R *is a regular language.*

Proof. Take

$$V' = \{a' \mid a \in V\},$$

$$h_3 : V^* \to V'^*, \quad h_3(a) = a', \ a \in V,$$

$$h_2 : V^* \to (V \cup \{c\})^*, \quad h_2(a) = ac, \ a \in V, \ c \text{ a new symbol},$$

$$h_1 : (V \cup V')^* \to V^*, \quad h_1(a) = a, \ a \in V, \ h_1(a') = \lambda, \ a \in V,$$

$$R = \{aa' \mid a \in V\}^*,$$

$$U = \{c\}.$$

The morphism h_3 primes each symbol, h_2 marks each symbol with c, GSS replaces each occurrence of c (in a prefix) by a symbol, the intersection with R selects the strings obtained by replacing each c by a primed symbol associated to the left neighbour; finally, h_1 erases all primed symbols, thus leaving a string in $L_1 \cap L_2$. The equality in the lemma is obtained. □

Theorem 2.7. *The families of linear and of context-free languages are not closed under the operation GSS.*

Proof. These families are not closed under intersection. □

3. Decidability of thinness

The main results in this section concern context-free languages. We begin with a known lemma.

Lemma 3.1. *Given a context-free grammar* G, *an equivalent context-free grammar* G' *satisfying the following three conditions can be effectively constructed:* (i) *Every nonterminal of* G' *is reachable from the initial symbol* S. (ii) G' *has no chain productions* $A \to B$, *where* A *and* B *are nonterminals.* (iii) *Every nonterminal of* G', *with the possible exception of the initial symbol, generates infinitely many terminal words. Moreover, if* G *is unambiguous, so is* G'.

Proof. The well-known transformations do not introduce ambiguity. On the contrary, they may remove it. The possible exception in (iii) concerns the case where $L(G)$ is finite. □

Lemma 3.2. *Assume that G' is a nonlinear unambiguous grammar generating a nonempty language and satisfying the conditions of Lemma* 3.1. *Then $L(G')$ is not thin.*

Proof. It follows from the assumptions that $L(G')$ is infinite and that there is a derivation

$$S \overset{+}{\Rightarrow} x_1 A x_2 B x_3$$

according to G', where x_1, x_2, x_3 are terminal words and A, B are nonterminals. Since A generates infinitely many terminal words, it must generate a nonterminal generating itself. We assume, without loss of generality, that A and B are themselves such nonterminals. (This can be done because the additional terminal words possibly produced may be included in x_1, x_2, x_3 above.) Thus, for some terminal words y_1, y_2, y_3, y_4,

$$A \overset{+}{\Rightarrow} y_1 A y_2, \quad B \overset{+}{\Rightarrow} y_3 B y_4, \qquad |y_1 y_2| = p \geqslant 1, \ |y_3 y_4| = q \geqslant 1.$$

Let x_4 and x_5 be terminal words generated by A and B, respectively, and denote

$$|x_1 x_4 x_2 x_5 x_3| = r.$$

By pumping A $i \geqslant 0$ times and B $j \geqslant 0$ times, we see that the word

(*) $\quad x_1 y_1^i x_4 y_2^i x_2 y_3^j x_5 y_4^j x_3$

is in the language $L(G')$, the length of the word being

(**) $\quad ip + jq + r.$

Because G' is unambiguous, no two pairs (i, j) can give rise to the same word (*).

Consider now any integer $k \geqslant 1$. If we choose

$$i = nq, \qquad j = (k-n)p, \quad 0 \leqslant n \leqslant k,$$

each of the resulting $k+1$ choices gives in (**) the length $kpq + r$. Consequently, for any $k \geqslant 1$, $L(G')$ contains at least $k+1$ words of length $kpq + r$. (To prove Lemma 3.2, actually two words of the same unbounded length would suffice. We have given the result in a somewhat stronger form for the purposes of Section 5. The weaker form suffices for Theorem 3.5.) □

Lemma 3.3. *Assume that G' is a linear unambiguous grammar generating an infinite language and satisfying the conditions of Lemma* 3.1. *Assume further that there are nonterminals A and B and terminal words x_1, x_2, x_3, x_4 and y_1, y_2, y_3, y_4 such that we have a derivation*

(D) $\quad S \overset{*}{\Rightarrow} x_1 A x_2 \overset{+}{\Rightarrow} x_1 x_3 A x_4 x_2 \overset{*}{\Rightarrow} x_1 x_3 y_1 B y_2 x_4 x_2 \overset{+}{\Rightarrow} x_1 x_3 y_1 y_3 B y_4 y_2 x_4 x_2,$

where the productions applied in the two derivations

$$A \overset{*}{\Rightarrow} x_3 A x_4 \quad \text{and} \quad B \overset{*}{\Rightarrow} y_3 B y_4$$

are not exactly the same. Then $L(G')$ is not thin.

Proof. Observe that the assumption concerning the two derivations is made in order to exclude the case where the two pumping situations collapse into one. If they collapse, two different pumpings may still yield the same derivation. For instance, using only the production $S \to aSb$, a derivation (D) is obtained, where

$$x_1 = x_2 = y_1 = y_2 = \lambda, \qquad x_3 = y_3 = a, \qquad x_4 = y_4 = b.$$

Similarly, the productions

$$S \to aAb, \qquad A \to cSd$$

do not yield (D) satisfying the additional condition required. On the other hand, we obtain (D) as required with the two productions

$$S \to aSb \quad \text{and} \quad S \to abSa,$$

as well as with the three productions

$$S \to aAb, \qquad A \to cSd, \qquad A \to cA.$$

Lemma 3.3 can now be established in the same way as Lemma 3.2. We denote

$$|x_3 x_4| = p, \qquad |y_3 y_4| = q, \qquad |x_1 x_2 y_1 y_2| = r$$

and obtain, for any $k \geqslant 1$, $k+1$ words of the same length $kpq + r$. The unambiguity of G' and our assumption concerning (D) guarantee that these words are distinct. □

Lemma 3.3 holds for nonlinear unambiguous grammars as well.

We are now ready to establish our main decidability result.

Theorem 3.4. *It is decidable whether or not an unambiguous context-free grammar generates a thin language.*

Proof. We just construct for our language a grammar G' satisfying Lemma 3.1. If $L(G')$ is finite, it is thin. Otherwise, we may assume by Lemmas 3.2 and 3.3 that G' is an unambiguous linear grammar such that no derivation (D) satisfying the required condition exists according to G'.

Let us call a nonterminal A *looping* if it is possible to derive from A (in a positive number of steps) a word containing A. (Looping nonterminals are sometimes also called recursive.)

$$A \overset{+}{\Rightarrow} x_A A y_A$$

is unique, provided A does not occur in the intermediate steps. (Thus, we take the first word where A occurs.) This means that also the words x_A and y_A are unique. Denote $|x_A y_A| = p_A$. It follows that $p_A \geqslant 1$.

Consider terminal words w_A derivable from A in such a way that A does not occur in the intermediate steps of the derivation. By the nonexistence of (D), it follows that there are only finitely many such words w_A. Similarly, there are only finitely many pairs (u_A, v_A) such that

$$S \overset{+}{\Rightarrow} u_A A v_A,$$

and A does not occur in the intermediate steps. (Observe that if looping nonterminals occur either in a derivation of $u_A A v_A$ from S or in a derivation of w_A from A, then they have to belong to the unique loop determined by A. In fact, S itself may belong to this loop.)

Suppose there are two triples (u_A, w_A, v_A) and (u'_A, w'_A, v'_A) such that

$$|u_A w_A v_A| \equiv |u'_A w'_A v'_A| \pmod{p_A}.$$

Then we conclude that $L(G')$ is not thin. Otherwise, there are integers r with

$$0 \leqslant r_1 < \cdots < r_t < p_A, \quad t \geqslant 1,$$

such that G' generates by derivations via A exactly one word of each length from the residue classes determined by r_j, $1 \leqslant j \leqslant t$. Here some "initial mess" may have to be excluded because the lengths $|u_A w_A v_A|$ may exceed p_A. Specifically, for some n_A, the subset of $L(G')$, consisting of words longer than n_A and generated by derivations via A, contains exactly one word of length $n > n_A$ if

$$n = ip_A + r_j, \quad i \geqslant 0,\ 1 \leqslant j \leqslant t,$$

and no other words with length greater than n_A.

If G' contains a looping nonterminal B not in the unique loop determined by A, B is treated in the same way, yielding the period p_B and, in case nonthinness cannot be concluded immediately, the set of possible lengths n is expressed in terms of p_B and the residue indicators. In can be immediately decided whether or not the length sets corresponding to A and B intersect. If they do, $L(G')$ is not thin. If their intersection is empty, we continue the process until all looping nonterminals have been exhausted. The language $L(G')$ is thin if no intersection of the length sets is found.

Observe that the start symbol S is not looping if there are at least two loops. The best way to handle the situation in case of several loops is the following. One nonterminal is picked from each loop, and the corresponding period is computed. Let p be the least common multiple of the resulting periods. The possible lengths can be expressed as residue classes (mod p). Non-thinness means that the same residue class is obtained twice, either from two different periods or twice from the same period. This concludes the proof of Theorem 3.4. □

The use of unambiguity in the proof is quite essential to assure that the equally long words derived in different ways are, in fact, different. Languages over one letter are thin but our proofs do not capture the fact that the resulting words coincide. A somewhat less trivial example is the following. Consider the (ambiguous) grammar with the productions

$$S \to aAbB, \qquad A \to baA(ba)^3, \qquad A \to ba,$$

$$B \to (ab)^3B(ab)^2, \qquad B \to ab.$$

Following the proof of Lemma 3.2, we obtain

$$p=8, \qquad q=10, \qquad r=6.$$

For each k, we obtain $k+1$ differently derived words of length $80k+6$ but they all coincide with the word $(ab)^{40k+3}$.

The following two theorems are immediate consequences of Lemma 3.2 and Theorem 3.4.

Theorem 3.5. *There are no nonlinear unambiguous thin languages.*

Theorem 3.6. *It is decidable whether or not a regular language is thin.*

We conjecture that the decidability result of Theorem 3.4 can be extended to concern all context-free languages. The proof calls for an analysis in the combinatorics of words concerning the possibilities of the words to coincide.

We now turn to undecidability.

Theorem 3.7. *It is undecidable whether or not the intersection of two linear languages is thin.*

Proof. We apply reduction to the Post Correspondence Problem. Consider an instance

$$(x_1, \ldots, x_n), (y_1, \ldots, y_n), \quad x_i, y_i \in \{a, b\}^+,$$

and define two linear grammars G_x and G_y with the terminal alphabet $\{a, b, c\}$ by listing the productions:

$$G_x: S_1 \to x_i S_1 ba^i, \quad i=1, \ldots, n,$$

$$G_y: S_1 \to y_i S_1 ba^i, \quad i=1, \ldots, n,$$

$$G_x \text{ and } G_y: S \to S_2,\ S_2 \to cS_2,\ S_2 \to \lambda,\ S \to S_1,\ S_1 \to c.$$

(S is the start symbol in both grammars.)

Clearly, both $L(G_x)$ and $L(G_y)$ contain c^*. Their intersection contains other words exactly in case our instance of PCP possesses a solution. If it possesses a solution, it

possesses infinitely many of them since a solution can be repeated arbitrarily many times. Consequently, the intersection $L(G_x) \cap L(G_y)$ is thin exactly in case the instance possesses no solution. This completes the reduction and the proof. □

We say that a language is *co-thin* if its complement is thin.

Theorem 3.8. *It is undecidable whether or not a linear language is co-thin.*

Proof. We apply the proof of the Lemma 5.7 from [6]. Given an instance of PCP as in the preceding proof, we denote by L_x the subset of

$$L = \{a, b\}^* c \{ba, ba^2, \ldots, ba^n\}^*,$$

consisting of all words *not* of the form

$$x_{i_1} x_{i_2} \ldots x_{i_k} c b a^{i_k} \ldots b a^{i_1}.$$

The language L_y is defined similarly.

In [6] a linear grammar G_1 is constructed for $L_x \cup L_y$. Consider also the regular language

$$R = \sim L \cap \sim c^*.$$

Starting from G_1, a linear grammar G can be constructed such that

$$L(G) = L_x \cup L_y \cup R.$$

Observe that $L = L_x \cup L_y$ iff PCP has no solution. Moreover, each solution gives rise to a word in $L - (L_x \cup L_y)$. Since $\sim R = L \cup c^*$, we conclude that $L(G)$ is co-thin exactly in case no solution exists. This completes the proof. □

The last theorem in this section is an immediate corollary of either Theorem 3.7 or 3.8. We return to related matters in Section 5.

Theorem 3.9. *It is undecidable whether or not a given context-sensitive language is thin.*

4. Minimal length-complete subsets

Consider a fixed ordering of the alphabet. It induces a lexicographic ordering of the words. A natural MLC subset of a language L is the language $L_{\min}$, obtained by taking from all words of L of the same length only the first in the lexicographic ordering. If L belongs to a class of languages, it is natural to ask whether or not $L_{\min}$ is also in the class and, moreover, is it there effectively.

The next theorem is a corollary of Eilenberg's cross-section theorem [3]. We present a somewhat different proof.

Theorem 4.1. *For every regular language L, the language $L_{\min}$ is regular, and a regular grammar for it can be effectively constructed.*

Proof. Let $G=(V_N, V_T, S, P)$ be a regular grammar for L (hence, containing rules of the form $A \to aB$, $A \to a$, $A, B \in V_N$, $a \in V_T$, possibly also $S \to \lambda$ if $\lambda \in L$). Assume that V_T is totally ordered. We construct a regular grammar

$$G'=(V'_N, V_T, S', P')$$

for generating the set

$$L_M=\{x \in L \mid \text{there is } y \in L,\ |x|=|y|,\ y<x \text{ in the lexicographic order}\}.$$

Then, clearly,

$$L_{\min}=L-L_M.$$

As the family of regular languages is closed under difference, it will follow that also $L_{\min}$ is regular.

The components of G' are defined as follows:

$$V'_N=\{(A, B), [A, B] \mid A, B \in V_N\},$$

$$S'=(S, S),$$

$$P'=\{(A, B) \to a(A', B') \mid A \to aA' \in P,\ B \to aB' \in P\}$$

$$\cup \{(A, B) \to a \mid A \to a \in P,\ B \to b \in P,\ a>b\}$$

$$\cup \{(A, B) \to a[A', B'] \mid A \to aA' \in P,\ B \to bB' \in P,\ a>b\}$$

$$\cup \{[A, B] \to a[A', B'] \mid A \to aA' \in P,\ B \to bB' \in P,\ a, b \text{ arbitrary in } V_T\}$$

$$\cup \{[A, B] \to a \mid A \to a \in P,\ B \to b \in P,\ a, b \text{ arbitrary in } V_T\}.$$

If a terminal derivation in G' uses only symbols (A, B), then it produces a string $w=w_1 a$ for which a string $z=w_1 b$ is in $L(G)$, $a>b$. If a symbol $[A, B]$ is used, then the derivation produces a string $w=w_1 a w_2$ for which $z=w_1 b w_3$ is in $L(G)$, with $a>b$, w_2, w_3 arbitrary, but of the same length, Consequently, $L(G') \subseteq L_M$.

Conversely, let $x \in L_M$ be an arbitrary string and let x_0 be the string in L such that $|x_0|=|x|$, $x_0 \in L_{\min}$. Assume $x=x_1 a x_2$, $x_0=x_1 b x_3$, $x_1 \in V_T^*$, $a>b$, x_2, $x_3 \in V_T^*$, $|x_2|=|x_3|$. A derivation

$$(S, S) \overset{*}{\Rightarrow} x_1(A, B) \Rightarrow x_1 a[A', B'] \overset{*}{\Rightarrow} x_1 a x_2$$

is obtained in G', combining two derivations of x, x_0,

$$S \overset{*}{\Rightarrow} x_1 A \Rightarrow x_1 aA' \overset{*}{\Rightarrow} x_1 ax_2,$$

$$S \overset{*}{\Rightarrow} x_1 B \Rightarrow x_1 bB' \overset{*}{\Rightarrow} x_1 bx_3.$$

If $x_2 = x_3 = \lambda$, then

$$(S, S) \overset{*}{\Rightarrow} x_1 (A, B) \Rightarrow x_1 a$$

is possible in G' if

$$S \overset{*}{\Rightarrow} x_1 A \Rightarrow x_1 a,$$

$$S \overset{*}{\Rightarrow} x_1 B \Rightarrow x_1 b,$$

are derivations for x, x_0.

Consequently, $L_M \subseteq L(G')$, which finishes the proof. (Note that all the steps of the proof are effective.) □

Thus, the basic problems have been settled for regular languages: thinness is decidable, and, for every regular language, a regular MLC subset can be constructed. By Theorem 3.9, thinness of context-sensitive languages is undecidable. The next theorem settles the problem concerning MLC subsets. The proof can be carried out using linear bounded automata. We prefer a grammatical construction making use of the recent result concerning the closure of context-sensitive languages under complementation.

Theorem 4.2. *For every context-sensitive language L, the language $L_{\min}$ is effectively context-sensitive.*

Proof. We proceed in a similar way as in the proof of Theorem 4.1. Namely, given a context-sensitive grammar $G = (V_N, V_T, S, P)$ for L, construct a context-sensitive grammar G' for the language L_M of nonminimum strings in L. As the family of context-sensitive languages is closed under intersection [5] and complementation [8], it is also closed under difference; hence, $L_{\min} = L - L_M$ is also a context-sensitive language.

In order to obtain G', we first construct a type-0 grammar G'' working as follows:

(i) Start by constructing a string

$$X_1 ZSX_2 X_3 SX_4,$$

where X_1, X_2, X_3, X_4 are end markers and Z is a scanner;

(ii) using the rules in P, construct a string

$$X_1 Z w X_2 X_3 w' X_4$$

with $w, w' \in L(G)$;

(iii) using the scanner Z, move the markers X_1, X_3 to the right, passing over one terminal symbol at each step, synchronously; if at some stage we find that $w = w_1 a w_2$, $w' = w_1 b w_3$, $a > b$, then X_1, X_3 are replaced by Y_1, Y_3, respectively; also Y_1, Y_3 are moved to the right, passing simultaneously over one symbol; if they reach X_2, X_4, respectively, at the same time, that is, $|w| = |w'|$, then the string w' is erased and also the markers and the scanner are erased; if either X_1, X_3 reach X_2, X_4, respectively (hence, $w = w'$) or only one of X_1, X_3 reaches X_2, X_4 (hence, $|w| \neq |w'|$), then the derivation is blocked.

The details of this construction are left to the reader. It is clear that such a type-0 grammar G'' for L_M can be constructed. As the workspace of G'' is linearly bounded (in order to generate a string w, we have to erase a string w' with $|w'| = |w|$, as well as the other 5 symbols – markers and scanners), L_M is context-sensitive.

A context-sensitive grammar G' for L_M can be effectively constructed (see the proof of the workspace theorem in [5]); then a grammar for the complement of L_M can be effectively constructed (the proof in [8] is effective); finally, given two context-sensitive grammars, a grammar generating the intersection of their languages can be effectively constructed. In conclusion, a context-sensitive grammar for $L_{\min}$ can be effectively constructed.

As regards context-free languages, both basic problems remain open: decidability of thinness and the construction of an MLC subset. By Theorem 4.2, $L_{\min}$ is context-sensitive but it remains open whether or not it is context-free. For instance, for the Dyck language over the alphabet $\{a, b\}$ (known to be nonlinear), we have

$$D_{\min} = \{a^n b^n \mid n \geq 0\} \quad \text{or} \quad D_{\min} = (ab)^*,$$

depending upon whether $a < b$ or $b > a$. □

The last theorem in this section gives a partial result. The proof shows that the theorem could be stated, instead of context-free languages, for any family of languages closed under union and intersection with regular languages and containing no nonregular languages over one letter.

Theorem 4.3. *Assume that a context-free language L is given as the union of finitely many thin context-free languages. Then a thin MLC subset L' of L can be effectively constructed.*

Proof. Consider a context-free language $L \subseteq V^*$ which can be written as

$$L = \bigcup_{i=1}^{n} L_i,$$

with L_i as thin context-free languages, $1 \leqslant i \leqslant n$. Construct the languages $L_{k,j}$, L'_k, $0 \leqslant k \leqslant n-1$, $k+1 \leqslant j \leqslant n$, as follows:

(a) $L_{0,j} = L_j$, $1 \leqslant j \leqslant n$, $L'_0 = L_1$,

(b) $L_{k,j} = h^{-1}(h(L_{k-1,j}) - h(L'_{k-1})) \cap L_{k-1,j}$, $k+1 \leqslant j \leqslant n$, $L'_k = L'_{k-1} \cup L_{k,k+1}$

for $k \geqslant 1$, where $h: V^* \to a^*$ is the morphism defined by $h(c) = a$ for all $c \in V$.
Then take $L' = L'_{n-1}$. This is the language we are looking for.

Claim 1. *L' is context-free.*

In fact, all languages $L_{k,j}$, L'_k, $0 \leqslant k \leqslant n-1$, $k+1 \leqslant j \leqslant n$, are context-free. For $k=0$ the assertion is trivial. Then L'_k is the union of the context-free language L'_{k-1} (the induction hypothesis) and $L_{k,k+1}$, which is also context-free: it is the intersection of the context-free language $L_{k-1,k+1}$ with a regular language.

Claim 2. *L' is thin.*

In fact, all languages L'_k, $0 \leqslant k \leqslant n-1$, are thin. For $k=0$ the assertion is trivial. Assume that some L'_k, $k \geqslant 1$, is not thin. As $L'_k = L'_{k-1} \cup L_{k,k+1}$, it follows that either $L_{k,k+1}$ is not thin or there are $x \in L'_{k-1}$, $y \in L_{k,k+1}$, with $|x| = |y|$. The former case is impossible, as $L_{k,k+1}$ is a subset of L_{k+1}. In the latter case, we must have $y \in L_{k,k+1}$; as $h(x) = h(y)$ and $x \in L'_{k-1}$, we have $h(y) \notin h(L_{k-1,k+1}) - h(L'_{k-1})$, which implies $y \notin L_{k,k+1}$, contradiction.

All L'_k are thin; hence, also $L' = L'_{n-1}$ is thin.

Claim 3. *L' is a length-complete subset of L.*

Clearly, L' is a subset of L. Take an integer m such that $L \cap V^m \neq \emptyset$. Let i_0 be the smallest i such that $L_i \cap V^m \neq \emptyset$, $1 \leqslant i \leqslant n$. We have $L_{i_0} \cap V^m = \{x\}$ as L_{i_0} is thin. No language L_j, $j < i_0$, contains a string y with $|y| = |x|$; hence, no language $L_{k,j}$, $0 \leqslant k \leqslant j-1$, $1 \leqslant j < i_0$, contains such a string y. This means, also L'_k, $0 \leqslant k \leqslant i_0 - 2$, do not contain strings y with $|y| = |x|$. Consequently, $x \in L'_{i_0-1}$; as $L'_k \subseteq L'_{k+1}$ for all k, we have $x \in L'_{n-1}$; hence $L' = L'_{n-1}$ contains a string of length m, that is, it is length-complete.

5. k-thinness and slenderness

We now introduce a natural extension of the notion of thinness.

For $k \geqslant 1$, a language L over the alphabet V is called *k-thin* if, for almost all n,

$$\text{card}\,(L \cap V^n) \leqslant k.$$

(Thus, a language is thin iff it is 1-thin.) The language L is *slender* if it is k-thin, for some k.

The theory developed in Section 3 can be extended to k-thinness and slenderness. In fact, some arguments were formulated already in Section 3 to cope with the extension.

Lemma 5.1. *Lemmas* 3.2 *and* 3.3 *hold true also with the final conclusion "then* $L(G)$ *is not slender".*

Proof. It was shown in the proofs that, for any k, $L(G')$ contains $k+1$ words of the same length. Hence, $L(G')$ cannot be slender.

Theorem 5.2. *For each* k, *the* k*-thinness of an unambiguous context-free language is decidable. The slenderness of an unambiguous context-free language is decidable.*

Proof. The cases covered by Lemma 5.1 can be excluded. The conclusion is that $L(G')$ is not slender and, hence, not k-thin for any k. □

The remaining case is that of a linear grammar with the special property discussed in the proof of Theorem 3.4. In this case $L(G')$ is always slender. The k-thinness is decided by the same counting argument as in Theorem 3.4; now instead of two words of the same length, we try to avoid $k+1$ words of the same length.

The corollaries are obtained as in Section 3.

Theorem 5.3. *There are no nonlinear unambiguous slender languages.*

Theorem 5.4. *For each* k, *the* k*-thinness of a regular language is decidable. The slenderness of a regular language is decidable.*

We now turn to undecidability.

Theorem 5.5. *For each* k, *the* k*-thinness of a context-sensitive language is undecidable. The slenderness of a context-sensitive language is undecidable.*

The proof of Theorem 5.5 is omitted, the argument being straightforward. First two candidate words for PCP are generated next to each other, for instance, using grammars G_x and G_y of Theorem 3.7. A scanner then checks whether the candidates yield a solution. If they do, a grammar generating a language violating k-thinness is entered.

Theorems 3.7 and 3.8 can be extended to the undecidability of k-thinness for any given k. For this purpose, it suffices to introduce several copies of the alphabets. However, the arguments cannot be directly extended to concern the undecidability of slenderness.

The next natural step in the extension of the notion of thinness would be to bound the number of words of the same length m by a linear function of n. Such considerations, also related to the density of a language, lie outside the scope of this paper.

We conclude this section with the following result, certainly interesting on its own right, obtained as a consequence of Lemma 5.1.

Consider the minimal finite deterministic automaton A accepting a regular language R. We say that R possesses the *unique loop property* if, whenever s is a state of A appearing twice in a path leading from the initial state to one of the final states, each of the following three conditions is satisfied: (i) There are only finitely many paths from the initial state to s which do not contain s as an intermediate node. (ii) There are only finitely many paths from s to one of the final nodes which do not contain s as one of the intermediate nodes. (iii) All words taking A from s to s are powers of a unique nonempty word.

Theorem 5.6. *A regular language R is slender if and only if R possesses the unique loop property. For every polynomial $p(x)$, there is a slender regular language R that is not k-thin for any $k \leqslant p(\mathrm{ind}(R))$, where $\mathrm{ind}(R)$ is the index of R.*

Proof. The first sentence follows by our previous arguments. To prove the second sentence, denote by $W(m)$ the set of all words of length at most m over the alphabet $\{a_1, \ldots, a_r\}$, $m \geqslant 1$, $r \geqslant 2$. Then the index of the language

$$R(m) = a_1^* W(m)$$

equals $m+2$, whereas $R(m)$ is not k-thin for any $k < r^m$.

6. Conclusion

The main open problems mentioned above concern the extension of our results to cover all context-free languages. Also a more detailed study of the cryptographic aspects, especially from the point of view of public-key systems, as well as the thinness of languages generated by L systems, constitute interesting research areas. The latter is related to some celebrated open problems, such as the decidability of the existence of 0 in a Z-rational sequence. We hope to return to these issues.

Acknowledgment

We thank the anonymous referee for useful comments.

References

[1] S. Ginsburg, *Algebraic and Automata-Theoretic Properties of Formal Languages* (North-Holland, Amsterdam, 1975).
[2] Gh. Păun, *Grammars for Economic Processes* (Tehnică, București, 1980) (in Romanian).
[3] J. Sakarovitch, Deux remarques sur un théorème d'Eilenberg, RAIRO. *Theoret. Comput. Sci.* **17** (1983) 23–48.
[4] A. Salomaa, *Public-Key Cryptography* (Springer, Berlin, 1990).
[5] A. Salomaa, *Formal Languages* (Academic Press, New York, 1973).
[6] A. Salomaa, *Jewels of Formal Language Theory* (Computer Science Press, Rockville, 1981).
[7] L. Sântean, On insertion and deletion in formal languages, Ph.D. Dissertation, Turku University 1991.
[8] R. Szelepcsényi, The method of forcing for nondeterministic automata, *Bull. EATCS* **33** (1987) 96–100.