Acta Cybernetica **15** (2002) 339–351.

Factorizations of Languages and Commutativity Conditions *

Alexandru Mateescu[†], Arto Salomaa[‡], and Sheng Yu[§]

Abstract

Representations of languages as a product (catenation) of languages are investigated, where the factor languages are "prime", that is, cannot be decomposed further in a nontrivial manner. In general, such prime decompositions do not necessarily exist. If they exist, they are not necessarily unique - the number of factors can vary even exponentially. The paper investigates prime decompositions, as well as the commuting of the factors, especially for the case of finite languages. In particular, a technique about commuting is developed in Section 4, where the factorization of languages L_1 and L_2 is discussed under the assumption $L_1L_2 = L_2L_1$.

Keywords: finite language, catenation, commutativity of languages, prime decomposition

1 Introduction

Prime factorizations of natural numbers and their uniqueness constitute one of the really fundamental issues in all mathematical sciences. On the other hand, in the theory of formal languages, the operation of *product* or *catenation* was introduced already at a very early stage. Clearly, any language L can be expressed as a product of itself and the language $\{\lambda\}$ consisting of the empty word λ. We refer to such decompositions of L as *trivial*, and say that $L \neq \{\lambda\}$ is *prime* if it has only trivial decompositions. In a *prime decomposition* for a language L every factor is a prime. Although questions dealing with primality can be viewed as fundamental in language theory, rather little work in this area has been done so far, see, for instance, [10, 6]. [2] is an early reference dealing with finite languages. [7] develops

*This work has been partially supported by the Project 137358 of the Academy of Finland and by the Natural Sciences and Engineering Research Council of Canada grants OGP0041630. All correspondence to Sheng Yu.

[†]Faculty of Mathematics, University of Bucharest Academiei, 14, Bucharest, Romania E-mail: alexmate@pcnet.pcnet.ro

[‡]Turku Centre for Computer Science (TUCS) Lemminkäisenkatu 14A, 20520 Turku, Finland E-mail: asalomaa@utu.fi

[§]Department of Computer Science, University of Western Ontario London, Ontario, Canada N6A 5B7 E-mail: syu@csd.uwo.ca

a method according to which one may construct, with the maximal use of the distributive law, for every finite language F an expression from which the number of states and final states in the minimal deterministic automaton for F can be immediately seen. [10] contains results about the commuting of two languages in some special cases.

The following remarks about related papers are in order. A systematic study about decompositions was initiated in the technical report [5]. This paper is the "journal version" of the report [5], while [9] is the "conference version" of it. The report [5] has given impetus to further research, for instance, [3, 4]. We have included in this paper material from [9] only insofar it increases readability. In particular, our main technical contribution in this paper, Section 4, is disjoint from [9].

We begin with the following basic observation about finite languages. Whenever a nonempty finite language F can be written as a product

$$F = F_1 F_2 \dots F_k,$$

where none of the factors F_i, $1 \leq i \leq k$, is trivial, then k cannot be larger than the length of the longest word in F. Consequently, we have always a complete control of all possible decompositions, at least in principle. This does not hold true for infinite regular languages, where there is no bound for the number of factors. Still decompositions such as

$$\Sigma^* = L_1 L_2 = (\lambda + \Sigma + \Sigma^2 + \dots \Sigma^{n-1})(\Sigma^n)^*, n \geq 2,$$

convey definite information about Σ^*. (Here, as frequently in the sequel, "+" stands for union.) Indeed, they were instrumental in the proof for the fact that equations between regular expressions possess no finite basis, see [7] for details.

Every finite language (different from $\{\lambda\}$) possesses a prime decomposition. This follows by an obvious induction on the length of the longest word in the language. This is not true for infinite languages. For instance, no star language L ($L = K^*$, for some K) can possess a prime decomposition. Indeed, for infinite languages, decompositions other than prime decompositions can sometimes be quite useful. For instance consider the language L over the one-letter alphabet $\{a\}$,

$$L = \{a^i \mid i = 10, 13, 16, 17, 19, 20 \text{ or } i \geq 22\}.$$

L possesses a decomposition $L = L_1 L_2$, where $L_1 = (a^3)^+$ and $L_2 = (a^7)^+$. Here we definitely have a simplification of the original language, presented as a product of languages, although the factors are not prime. For instance, the total number of states in the minimal automata for L_1 and L_2 is much smaller than the number of states in the minimal automaton of L. Using the same idea and allowing an arbitrary number of factors, one can show that the number of states may grow exponentially in the transition from the decomposition to the original language. Somewhat similar matters are discussed also in Section 3. We hope to return to the discussion of this and other similar problems (which lie outside the scope of the present paper) in another context.

A brief description of the contents of the present paper follows. The reader is expected to be familiar with the very basics of formal languages and finite automata. One of the references [6, 7, 8] may be consulted if need arises.

Basic decidability results are presented in Section 2. They lead also to a notion very central in the study of regular languages, that of a *decomposition set*, originally introduced in [9]. Sections 3-5 deal exclusively with finite languages. In Section 3, we discuss decompositions of different lengths, as well as the testing of the primality of a finite language, also from the point of view of complexity. The two final sections deal with the *commuting* of two finite languages F_1 and F_2, that is, the validity of the equation $F_1F_2 = F_2F_1$. While this is a tricky problem in the general case, some special cases can be handled.

Our main results are contained in Section 4, where factorization of languages F_1 and F_2 is discussed under the assumption $F_1F_2 = F_2F_1$. Also a very efficient construction is presented in the case where one of the two languages involved is a singleton. The construction could be applicable also in other similar situations. The final Section 5 discusses some recent results and open problems.

It has not escaped our notice, especially in view of the many possible interpretations of finite languages and the central theoretical role of the problems studied in this paper, that the problems might turn out to be significant in certain applications. For instance, succinct representations of DNA nucleotide sequences certainly fall within this category. However, we have had no specific applications in mind.

2 Decomposition sets and decision problems

The notions of a *prime language* and a *prime decomposition* of a language were already defined in the Introduction. According to the definition, the language $\{\lambda\}$ consisting of the empty word is not prime. Thus, all factor languages in a prime decomposition are nontrivial. Depending on the language, the prime decomposition may be unique or there may be several prime decompositions for the same language. It is also possible that a language has no prime decompositions. However, every finite language possesses a prime decomposition.

Typical problems concerning the decomposition of finite languages are the following:

1. Is a given finite language prime?

2. Find all prime decompositions of a given finite language.

3. Find, for a given finite language, a prime decomposition possessing a specific property. (We might require, for instance, that the total number of states in the automata accepting the prime factors is minimal.)

It is obvious that all problems of this nature are decidable for finite languages. The complexity issues lie mainly outside the scope of this paper. In many cases, an exhaustive search is the only algorithm we know for a specific problem.

We now present some simple examples, due to [9], of prime and nonprime finite langauges. Consider the languages over the alphabet $\{a\}$, defined by

$$F_{n,k} = \lambda + a^k + a^{2k} + \cdots + a^{nk}, n \geq 2, k \geq 1,$$

$$F'_n = \lambda + a^2 + a^3 + a^4 + \cdots + a^n, n \geq 4.$$

Let, further, $F''_n, n \geq 4$, denote any language consisting of λ and a^n and, in addition, of arbitrarily many words a^i with $n/2 \leq i \leq n$. Then no language $F_{n,k}$ is prime, whereas all languages F''_n are prime. The language F'_n is prime iff $n = 4$.

Sometimes a slight change in a prime language induces a possibility for a decomposition. Consider the following two languages:

$$F = adba + acbb + bcaa + bdab,$$

$$F' = adba + adbb + bdaa + bdab.$$

Thus F' results from F by replacing the two occurrences of the letter c by the letter d. Then the language F is prime, whereas F' possesses the decomposition $F' = (adb + bda)(a + b)$. See [9] for details, as well as for the proof of the following theorem and for related references.

Theorem 1 *There is no algorithm for deciding whether or not a given linear language is prime. Consequently, the problem of primality is undecidable for context-free languages.*

The proof of Theorem 1 does not work for regular languages. Indeed, as Theorem 2.2 below shows, the primality problem is decidable for regular languages. We now recall from [9] a notion very suitable for the study of decompositions of regular languages. It is closely related to left quotients of regular languages. It shows how an arbitrary decomposition can be extended to one of finitely many specific decompositions, obtainable in a standard way.

Let R be a regular language over an alphabet Σ, and let $A = (Q, \Sigma, \delta, q_0, Q_F)$ be the minimal finite deterministic automaton for R. (Here Q is the set of states, q_0 the initial state, Q_F the set of final states, and δ the transition function. We extend δ to words over Σ. Thus, $\delta(q, w) = q'$ means that the word w takes A from the state q to the state q'.) For a nonempty subset $P \subseteq Q$, we consider the following two languages:

$$R_1^P = \{w \mid \delta(q_0, w) \in P\},$$

$$R_2^P = \bigcap_{p \in P} \{w \mid \delta(p, w) \in Q_F\}.$$

Lemma 1 *Let R and A be defined as above. Assume that $R = L_1 L_2$, where L_1 and L_2 are arbitrary languages. Define $P \subseteq Q$ by*

$$P = \{p \in Q \mid \delta(q_0, w) = p, \text{ for some } w \in L_1\}.$$

Then $R = R_1^P R_2^P$ and, moreover, $L_i \subseteq R_i^P$ for $i = 1, 2$.

Lemma 1 was established in [9]. Observe that the languages L_1 and L_2 above are quite arbitrary; they need not even be recursively enumerable. They can always be extended, without losing the validity of the decomposition, to regular languages obtainable from the minimal automaton for A. These resulting "standard" decompositions can always be expressed in terms of a *decomposition set.*

By definition, a nonempty subset $P \subseteq Q$ is a *decomposition set* (for a regular language R) if $R = R_1^P R_2^P$. The decomposition $R = R_1^P R_2^P$ referred to as the decomposition of R *induced* by the decomposition set P. We say that the decomposition $L = L_1 L_2$ of a language L is *included* in the decomposition $L = L_1' L_2'$ if $L_i \subseteq L_i'$, $i = 1, 2$. See [9] for the proof of the following result.

Theorem 2 *Every decomposition of a regular language R is included in a decomposition of R induced by a decomposition set. The problem of primality is decidable for regular languages.*

The algorithm obtained by checking through all possible decomposition sets is clearly exponential. It is likely that primality testing is NP-complete even for finite languages. Observe also that the decomposition induced by a decomposition set may be trivial. Indeed, we have $R_1^P = \{\lambda\}$ iff $P = \{q_0\}$ and q_0 has no incoming arrows. Similarly, $R_2^P = \{\lambda\}$ exactly in case $P = Q_F$ and λ is the only word taking A from each of the final states to a final state. Also the following result is an immediate corollary of Lemma 1.

Theorem 3 *Whenever a regular language has a nontrivial decomposition, it has a nontrivial decomposition where the factors are regular languages.*

We conclude this section with two open problems.

Open problem. Instead of catenation, we may take the *shuffle* operation to be the *product* operation for languages. Decompositions and primality can be defined for this product as well. Is the last sentence of Theorem 2 valid also now? In other words, is the primality of regular languages with respect to the shuffle product decidable? Although we have been able to settle some special cases, the case of an arbitrary regular language seems to be very tricky.

Open problem. Does Theorem 3 hold with "regular" replaced by "context-free"? It would be very strange to have an example of a context-free language L having nontrivial decompositions $L = L_1 L_2$, in all of which at least one of the languages L_1 and L_2 is non-context-free.

3 Primality testing

In the remainder of this paper we discuss only finite languages. A given finite language may possess several prime decompositions. It may even happen that two prime decompositions of the same language have no common factors. For instance,

$$(\lambda + a^2)(\lambda + a^2 + a^3 + a^4) = (\lambda + a^2 + a^3)^2,$$

where all factors are prime languages. Even the number of factors may vary drastically in different prime decompositions of the same language. The following contribution to Problem 2 of the preceding section was established in [9].

Theorem 4 *There are finite languages L_n having two prime decompositions with $O(n)$ and $O(\log n)$ factors.*

Theorem 4 was established in [9] using the following example. Consider numbers $n = 2^k$, $k \geq 1$, and languages

$$L_n = \lambda + a + a^2 + \cdots + a^{n-1}.$$

Then

$$L_n = (\lambda + a)^{n-1} = (\lambda + a)(\lambda + a^2)(\lambda + a^4)\ldots(\lambda + a^{2^{k-1}}).$$

The most straightforward examples about factorizations not unique are obtained in terms of languages over one-letter alphabet $\{a\}$. Other examples are easy to construct. For instance,

$$F' = \lambda + a + b + ab + b^2 + ab^2 + b^3 + ab^3 + b^4 + ab^4 = (\lambda + a + b + b^2 + ab^2)(\lambda + b)^2 =$$
$$= (\lambda + a + ab + b^2 + ab^2)(\lambda + b)^2 = (\lambda + a)(\lambda + b^2)(\lambda + b)^2,$$

where all languages within parentheses are primes.

Consider primality testing, Problem 1 mentioned in Section 2. There seems to be no other general method than trying all possible factors. Of course, in special instances, ad hoc arguments can be used to exclude factors of certain types. A special case consists of testing the primality of languages of the form

$$\lambda + a^{i_1} + a^{i_2} + \cdots + a^{i_n}, \tag{1}$$

where the i's are distinct positive integers. In this case primality testing can be reduced to a problem concerning sets of nonnegative integers as follows.

Let N be a set of nonnegative integers. We say that N has the *decomposition property* if there are nonempty subsets N_1 and N_2 of N, maybe overlapping or identical but both containing at least two elements, such that

$$N = \{n_1 + n_2 \mid n_1 \in N_1 \text{ and } n_2 \in N_2\}.$$

We also say that N *decomposes into* N_1 and N_2. (Recall here also the one-letter language L presented in the Introduction.)

Clearly, N can have the decomposition property only if $0 \in N$, in which case 0 belongs also to both N_1 and N_2. The following result is now obvious.

Lemma 2 *The language $L_N = \Sigma_{i \in N} a^i$ is prime iff the set N contains 0 and has not the decomposition property. More specifically, if N decomposes into N_1 and N_2 then*

$$L_N = (\lambda + \sum_{i \in N_1} a^i)(\lambda + \sum_{i \in N_2} a^i)$$

Although the problem of N possessing the decomposition property bears some resemblance to the subset sum problems, we have not been able to establish its NP-completeness. Of course, testing the primality of the languages (1) is only a special case of the general problem.

If c is a letter not in the alphabet of F, then $F + c$ is always prime. One can affect the same change also without introducing new letters.

Theorem 5 *Let F be a finite language whose minimal alphabet Σ contains at least two letters. Then for some $w \in \Sigma^+$, $F + w$ is prime.*

Proof. Let k be the length of the longest word in F. Let w be any word of length $2k+1$ such that there is a word in F whose first (resp. last) letter differs from the first (resp. last) letter of w. This requirement can be satisfied since Σ contains at least two letters. We claim that $F + w$ is prime. Assume the contrary: $F + w$ has a nontrivial decomposition $F + w = F_1F_2$. We can write $F_1 = F_1' + w_1$, $F_2 = F_2' + w_2$, $w = w_1w_2$. (Possibly F_i' is empty or $w_i = \lambda$.) One of the words w_1 and w_2 is of length greater than k. Assume that $|w_2| > k$. Then $F_1' = \emptyset$ because, if $x \in F_1'$, the word xw_2 is not in $F + w$. Thus, $F + w = w_1F_2$. But this is not possible because F contains a word whose first letter differs from the first letter of w_1. ($w_1 = \lambda$ would yield a trivial decomposition.) If $|w_1| > k$, we obtain similarly a contradiction, using the fact that F contains a word whose last letter differs from the last letter of w_2. This completes the proof. □

Theorem 5 can be extended to concern languages F over $\{a\}$ containing the empty word.

4 Factorization versus commutativity conditions

It was one of the very early results on combinatorics on words that two words u and v *commute*, $uv = vu$, iff both u and v are powers of the same word. No similar result is known for finite languages. When do two finite languages F_1 and F_2 commute, $F_1F_2 = F_2F_1$? We begin with the special case, where one of the languages is a singleton. The technique presented in this section, interesting also on its own right, shows in detail the structure of the two languages.

The following results are well known and can be found in, e.g., [6] or [10].

Lemma 3 *If $uv = vz$, $u, v, z \in \Sigma^*$, and $u \neq \lambda$, then $u = xy$, $v = (xy)^k x$, and $z = yx$ for some $x, y \in \Sigma^*$ and $k \geq 0$.*

Lemma 4 *If $uv = vu$, then there exists $x \in \Sigma^*$ such that $u = x^s$ and $v = x^t$ for some $s, t \geq 0$.*

Lemma 5 *If $u^m = v^n$ and $m, n \geq 1$, then $u = x^s$ and $v = x^t$ for some $x \in \Sigma^*$ and $s, t \geq 1$.*

Theorem 6 *Let $x \in \Sigma^*$ and $L \subseteq \Sigma^*$ be a finite language. If $xL = Lx$, then there exists $w \in \Sigma^*$ such that $x = w^s$ and $L = \bigcup_{i=1}^{n} \{w^{t_i}\}$, for $s, n, t_1, \ldots, t_n \geq 0$.*

Proof. The theorem holds trivially if $L = \emptyset$. If $L = \{y\}$, then $xy = yx$. By Lemma 4, we have $x = w^s$ and $y = w^t$ for some $s, t \geq 0$. Thus, the theorem holds.

Assume that the theorem holds for $L = \{y_1, \ldots, y_t\}$, $t < n$.

Now we consider the case when $t = n$, i.e., $L = \{y_1, \ldots, y_n\}$. We have the following three cases:

Case I. $xy_n = y_n x$. Then, by Lemma 4, $x = w_0^{s_0}$ and $y_n = w_0^{t_0}$ for some $w_0 \in \Sigma^*$ and $s_0, t_0 \geq 0$. Let $L' = \{y_1, \ldots, y_{n-1}\}$. Then $xL' = L'x$ since $xL = Lx$, $xy_n = y_n x$, and $xy_n \notin xL'$ and $y_n x \notin L'x$. By the induction hypothesis, $x = w_1^{s_1}$ and $L' = \bigcup_{i=1}^{n-1} \{w_1^{t_i}\}$ for some $w_1 \in \Sigma^*$ and $s_1, t_1 \geq 0$. Since $x = w_0^{s_0} = w_1^{s_1}$, w_0 and w_1 are powers of a common word w, i.e., $w_0 = w^l$ and $w_1 = w^m$. Then $x = w^{ls_0}$ and $L = \{w^{mt_1}, \ldots, w^{mt_{n-1}}, w^{lt_0}\}$. The theorem holds.

Case II. $xy_n \neq y_n x$. Then $xy_n = y_{i_1} x$ for some $i_1 \in \{1, \ldots, n-1\}$. If $xy_{i_1} = y_n x$, then let $L_1 = \{y_{i_1}, y_n\}$ and $L_2 = L - L_1$. Otherwise, $xy_{i_1} = y_{i_2} x$ for some $i_2 \in \{1, \ldots, n-1\} - \{i_1\}$. We continue this way until we get $xy_{i_m} = y_n x$, i.e.

$$xy_n = y_{i_1} x, \quad xy_{i_1} = y_{i_2} x, \quad \ldots\ldots, \quad xy_{i_m} = y_n x.$$

Consider the case $m < n - 1$. Let $L_1 = \{y_{i_1}, \ldots, y_{i_m}, y_n\}$ and $L_2 = L - L_1$. Then $xL_1 = L_1 x$ and $xL_2 = L_2 x$. By the induction hypothesis, we have

$$x = u^{s_1}, \quad L_1 = \bigcup_{i=1}^{m} \{u^{t_i}\}, \text{ and } x = v^{s_2}, \quad L_2 = \bigcup_{j=1}^{n} \{v^{t_j}\}.$$

Since $u^{s_1} = v^{s_2} = x$, we have $u = w^k$ and $v = w^l$ for $w \in \Sigma^*$ and $k, l \geq 0$. Therefore,

$$x = w^{ks_1}, \quad L = (\bigcup_{i=1}^{m} \{w^{kt_i}\}) \cup (\bigcup_{j=1}^{n} \{w^{lt_j}\}).$$

Case III. This case is the same as Case II except that $m = n - 1$, i.e, we have $xy_n \neq y_n x$ and

$$xy_n = y_{i_1} x, \quad xy_{i_1} = y_{i_2} x, \ldots\ldots, \quad xy_{i_{n-1}} = y_n x.$$

Since $xy_n = y_{i_1} x$ and $xy_{i_1} = y_{i_2} x$, we have, by Lemma 3,

$$x = (u_1 v_1)^{k_1} u_1, \quad y_n = v_1 u_1, \quad y_{i_1} = u_1 v_1$$

$$x = (u_2 v_2)^{k_2} u_2, \quad y_{i_1} = v_2 u_2, \quad y_{i_2} = u_2 v_2.$$

So, we have

$$(u_1 v_1)^{k_1} u_1 u_1 v_1 = u_2 v_2 (u_1 v_1)^{k_1} u_1.$$

Then, $u_1 v_1 = v_1 u_1$. Thus, u_1 and v_1 are powers of the same word $w_1 \in \Sigma^*$. So, $x = w_1^{s_1}$ and $y_1 = w_1^{t_1}$ for $s_1, t_1 \geq 0$. Similarly, we can show that, for $1 \leq i \leq n$,

$$x = w_i^{s_i} \text{ and } y_i = w_i^{t_i}.$$

Since $w_1^{s_1} = \ldots = w_n^{s_n} = x$, we know that $w_1, \ldots, w_n$ are powers of a common word w, i.e., $w_1 = w^{l_1}, \ldots, w_n = w^{l_n}$ by Lemma 5. Thus, $L = \bigcup_{i=1}^{n} \{w^{l_i t_i}\}$ and $x = w^{l_1 s_1}$. □

Let p and q be two natural numbers such that $(p,q) = 1$ and $p < q$. Define $N_p = \{1, \ldots, p\}$ and $N_q = \{1, \ldots, q\}$. Also define a function $\sigma : N_q \to N_q$ by $\sigma(i) = ((i+p-1) \bmod q)+1$. Thus, $\sigma(i) = i+p$ where the least positive remainder of the sum modulo q is taken. Since $(p,q) = 1$, it is clear that, for any $i \in N_q$, we have $\{i, \sigma(i), \ldots, \sigma^{q-1}(i)\} = N_q$ and $\sigma^q(i) = i$.

Let $w \in \Sigma^{tm}$, $t, m > 0$, i.e., $w = x_1 x_2 \cdots x_t$ and $x_i \in \Sigma^m$, $1 \le i \le t$. Denote by $(w)_i^{(m)}$, $1 \le i \le t$, the substring x_i of w. When m is understood, we simply write $(w)_i$.

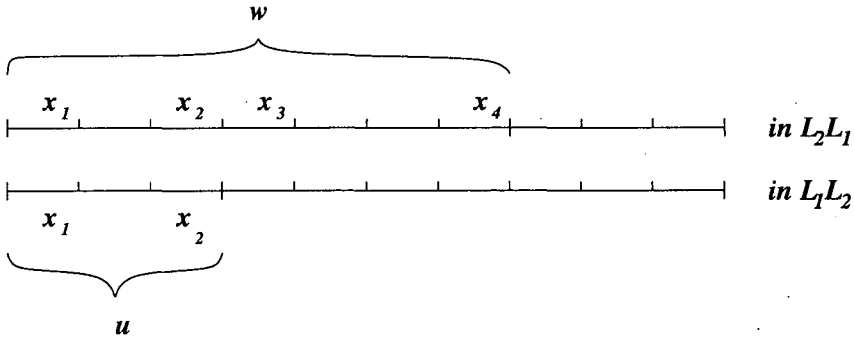
Let $L_1 \subseteq \Sigma^{pm}$, $L_2 \subseteq \Sigma^{qm}$, $p, q, m > 0$, $(p,q) = 1$, $p < q$, and $L_1 L_2 = L_2 L_1$. Then we have the following results.

Lemma 6 *Let $N_q' = \{i_1, \ldots, i_n\}$, $1 \le n \le q$, be a subset of N_q and $x_1, \ldots, x_n \in \Sigma^m$. If there exists $w \in L_2$ such that $(w)_{i_1} = x_1, \ldots, (w)_{i_n} = x_n$, then there exists $u \in L_1$ such that $(u)_{i_j} = x_j$ for all $i_j \in N_q' \cap N_p$.*

Proof. The lemma holds due to the facts that $L_1 L_2 = L_2 L_1$ and $p < q$. □

We explain this lemma by the following example.

Example 1 Let $p = 3$ and $q = 7$. Then $N_p = \{1,2,3\}$ and $N_q = \{1,2,\ldots,7\}$. Given $N_q' = \{1,3,4,7\}$ and $x_1, x_2, x_3, x_4 \in X = \Sigma^m$, there is $w \in L_2$ such that $(w)_1 = x_1$, $(w)_3 = x_2$, $(w)_4 = x_3$, and $(w)_7 = x_4$. Then, clearly, there is $u \in L_1$ such that $(u)_1 = x_1$ and $(u)_3 = x_2$, which is illustrated in the diagram below.

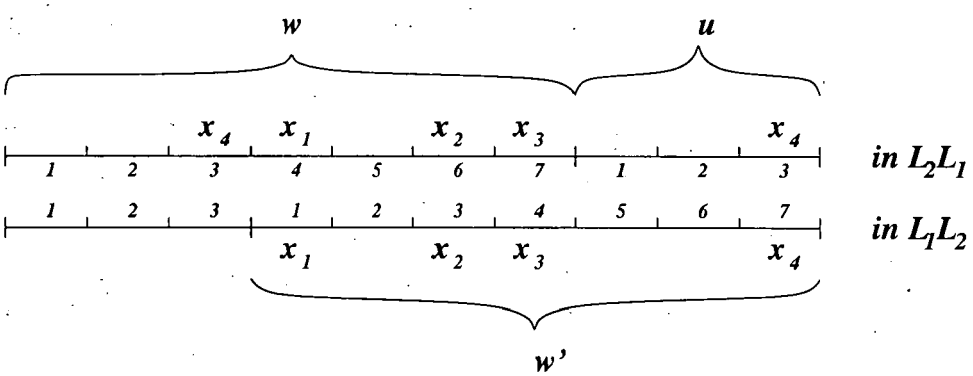


Lemma 7 *Let $N_q' = \{i_1, \ldots, i_n\}$, $1 \le n \le q$, be a subset of N_q and $x_1, \ldots, x_n \in \Sigma^m$. If there exists $w \in L_2$ such that $(w)_{\sigma(i_1)} = x_1, \ldots, (w)_{\sigma(i_n)} = x_n$, then there exists $w' \in L_2$ such that $(w')_{i_1} = x_1, \ldots, (w')_{i_n} = x_n$.*

Proof. Let $w \in L_2$ such that $(w)_{\sigma(i_j)} = x_j$, $i_j \in N'_q$. Then there exists $u \in L_1$ such that $(u)_{\sigma(i_j)} = x_j$ for all $\sigma(i_j) \in N'_q \cap N_p$ by Lemma 6. Since $wu \in L_2L_1$ and $L_2L_1 = L_1L_2$, we have $wu = vw'$ for $v \in L_1$ and $w' \in L_2$. Notice that $(w')_i = (w)_{\sigma(i)}$ for $1 \leq i \leq (q-p)$ and $(w')_i = (u)_{\sigma(i)}$ for $(q-p+1) \leq i \leq q$ due to the fact that $|v| = p$ and the definition of σ. Therefore, $(w')_{i_j} = (w)_{\sigma(i_j)} = x_j$, $1 \leq j \leq n$. □

We explain the above lemma and its proof by the following example.

Example 2 As in the previous example, let $p = 3$ and $q = 7$. Then $N_p = \{1, 2, 3\}$ and $N_q = \{1, 2, \ldots, 7\}$. We are given $N'_q = \{1, 3, 4, 7\}$ and $x_1, x_2, x_3, x_4 \in X = \Sigma^m$, and we know that there is $w \in L_2$ such that $(w)_{\sigma(1)} = x_1$, $(w)_{\sigma(3)} = x_2$, $(w)_{\sigma(4)} = x_3$, and $(w)_{\sigma(7)} = x_4$ (i.e., $(w)_4 = x_1$, $(w)_6 = x_2$, $(w)_7 = x_3$, and $(w)_3 = x_4$). Then, there is $u \in L_1$ such that $(u)_3 = x_4$ by Lemma 6. Since $wu \in L_2L_1 = L_1L_2$, we have $wu = vw'$ for $v \in L_1$ and $w' \in L_2$. We can see from the diagram below that $(w')_1 = x_1$, $(w')_3 = x_2$, $(w')_4 = x_3$, and $(w')_7 = x_4$.



Corollary 1 *Let $N'_q = \{i_1, \ldots, i_n\}$, $1 \leq n \leq q$, be a subset of N_q and $x_1, \ldots, x_n \in \Sigma^m$. If there exists $w \in L_2$ such that $(w)_{\sigma^k(i_1)} = x_1, \ \ldots, \ (w)_{\sigma^k(i_n)} = x_n$, for some k, $0 \leq k \leq q-1$, then there exists $w' \in L_2$ such that $(w')_{i_1} = x_1, \ldots, (w')_{i_n} = x_n$.*

Proof. Apply Lemma 7 k times. □

Theorem 7 *Let $L_1 \subseteq \Sigma^{pm}$, $L_2 \subseteq \Sigma^{qm}$, $p, q, m > 0$, $(p, q) = 1$, and $L_1L_2 = L_2L_1$. Furthermore, let $X = \{w \in \Sigma^m \mid wu \in L_1 \text{ for some } u \in \Sigma^*\}$. Then $L_1 = X^p$ and $L_2 = X^q$.*

Proof. First, we prove that $L_1 \subseteq X^p$ and $L_2 \subseteq X^q$. Define $Y_i = \{w \in \Sigma^m \mid uwv \in L_1 \text{ for } u \in \Sigma^{(i-1)m} \text{ and } v \in \Sigma^{(p-i)m}\}$, $1 \leq i \leq p$,

and $Z_j = \{w \in \Sigma^m \mid uwv \in L_2 \text{ for } u \in \Sigma^{(j-1)m} \text{ and } v \in \Sigma^{(q-j)m}\}$, $1 \leq j \leq q$. Then, clearly, $L_1 \subseteq Y_1 \cdots Y_p$ and $L_2 \subseteq Z_1 \cdots Z_q$. We know that $Y_1 = X$ and it is

obvious that $Z_1 = X$. Let $x \in Z_i$ for some i, $1 \leq i \leq q$. Then there is a word $w \in L_2$ such that $(w)_i = x$. It is clear that $i = \sigma^k(1)$ for some k, $0 \leq k \leq q-1$. Then, by Corollary 1, we know that there is $w' \in L_2$ such that $(w')_1 = x$. So, $x \in Z_1 = X$. Since x is chosen arbitrarily, we have shown that $Z_i \subseteq X$. Similarly, we can show that $Z_i \subseteq X$ for each i, $1 \leq i \leq q$. Therefore, we have $Z_1 \cdots Z_q \subseteq X^q$ and, thus, $L_2 \subseteq X^q$. As a consequence we have that $L_1 \subseteq X^p$.

Second, we prove that $X^p \subseteq L_1$ and $X^q \subseteq L_2$. In order to do so, we prove by induction on n, the cardinality of N'_q, $1 \leq n \leq q$, that for any $N'_q = \{i_1, \ldots, i_n\} \subseteq N_q$ and $x_1, \ldots, x_n \in X$, there exists $w \in L_2$ such that $(w)_{i_k} = x_k$, for $1 \leq k \leq n$. For $n = 1$, let $N'_q = \{i\}$ and x be an arbitrary word in X. If $i = 1$, it is clear that there exists $w \in L_2$ such that $(w)_1 = x$. Otherwise, there $\sigma^k(i) = 1$ for some k, $1 \leq k \leq q-1$. So, by Corollary 1, we have $w \in L_2$ and $(w)_i = x$.
For the induction step, let $N'_q = \{i_1, \ldots, i_n\}$ and $x_1, \ldots, x_n \in X$. Denote $N'_p = N'_q \cap N_p$. If $N'_p \neq \emptyset$ and $N'_q - N'_p \neq \emptyset$, then both $\#N'_p < n$ and $\#(N'_q - N'_p) < n$. Then there exist $u \in L_1$ such that $(u)_{i_k} = x_k$ for $i_k \in N'_p$ and $v \in L_2$ such that $(v)_{\sigma^{q-1}(i_l)} = x_l$ for $i_l \in N'_q - N'_p$ by the induction hypothesis. Since $L_1L_2 = L_2L_1$, we have $uv = wz$ for $w \in L_2$ and $z \in L_1$. Clearly, $(w)_{i_k} = x_k$ for all $i_k \in N'_q$. Otherwise ($N'_p = \emptyset$ or $N'_q - N'_p = \emptyset$), there is an integer $k > 0$ such that $\sigma^k(N'_q)$ satisfies the condition. Then by Corollary 1 and the above arguments, we have $w \in L_2$ such that $(w)_{i_k} = x_k$.

Let $n = q$, i.e., $N'_q = N_q$. Then we have proved that $X^q \subseteq L_2$. Using Lemma 6, we get $X^p \subseteq L_1$. Therefore, we have $L_1 = X^p$ and $L_2 = X^q$. □

5 Further results and open problems

One might be tempted to conjecture that two finite languages F_1 and F_2 commute, $F_1F_2 = F_2F_1$, iff there is a finite language F such that both F_1 and F_2 are unions of powers of F. (Indeed, such a conjecture was presented in [9].) Clearly, if both F_1 and F_2 are of the form

$$F^{i_1} + F^{i_2} + \ldots + F^{i_n},$$

where also $F^0 = \{\lambda\}$ can appear among the terms of the union, then $F_1F_2 = F_2F_1$.

However, the *converse* is not true: F_1 and F_2 may commute without being unions of powers of the same set. The examples used in connection with Theorem 4 can be applied to provide counterexamples. For instance, denote

$$L_1 = a + a^2 + a^3, \quad L_2 = a + a^3, \quad L_3 = \lambda + a.$$

Then

$$L_1L_3 = L_2L_3 (= L_3L_1 = L_3L_2).$$

If we now denote $F_i = L_i + L_3\{b\}L_3$, $i = 1, 2$, it follows that $F_1F_2 = F_2F_1$. It is also clear that F_1 and F_2 cannot be represented as unions of powers of the same

set. (First examples to this effect were given in [3], where it is also shown that the converse holds in case the cardinality of one of the sets F_1 and F_2 is at most 2.)

The validity of the converse, as well as the unique decomposition, can be directly established in some special cases.

For instance, let $\mathcal{E}$ consist of all nonempty finite languages F, where all words of F are of equal length. Then we get immediately the following result.

Lemma 8 *Assume that F is a language in $\mathcal{E}$ and that $F = F_1F_2$, for some F_1 and F_2. Then both F_1 and F_2 are in $\mathcal{E}$.*

Lemma 8 shows that the languages in $\mathcal{E}$ possess a unique prime decomposition and that $\mathcal{E}$ is a free monoid with respect to catenation. Observe that $\mathcal{E}$ is not finitely generated. See [9] for a more detailed discussion.

Thus, the equation $F_1F_2 = F_2F_1$ holds for languages in $\mathcal{E}$ only in case both F_1 and F_2 are powers of the same language X. Moreover, if one of the languages, say F_2, is an arbitrary finite language, we may present it as a (finite) union of languages in $\mathcal{E}$ and use the same argument to show the existence of a language X such that F_1 is a power of X and F_2 is a (finite) union of powers of X. This and other similar results have been established in [10].

The technique in the preceding section was based on more detailed arguments and yields a direct construction of the set X.

In conclusion, we present some general remarks and open problems concerning the *converse* mentioned above. What can be said, in general, about two commuting finite languages F_1 and F_2?

Open problem. Assume that $F_1F_2 = F_2F_1$ holds for two finite languages F_1 and F_2. Characterize the cases, where F_1 and F_2 are *not* unions of powers of the same language. In the sequel we refer to such cases as *exceptional.*

One possible approach to this problem is to consider *positive* decompositions, [9]. As seen above, the ambiguities caused by the presence of λ seem to be the reason behind exceptional cases.

Another approach is is to have an upper bound for the cardinality of one of the two finite languages, say F_1. We already mentioned that if F_1 is of cardinality at most 2 then, independently of F_2, the case is not exceptional, [3]. On the other hand, the example

$$F_1 = a + ab + ba + bb, \qquad F_2 = F_1 + F_1^2 + bab + bbb$$

given in [4] shows that the upper bound 4 for the cardinality of F_1 is not sufficient. It is an open problem whether or not the upper bound 3 is sufficient.

In our few final remarks about commuting, the languages considered are not necessarily finite. Following [4], we say that a finite language $F \subseteq \Sigma^*$ possesses the *Bergman type characterization, BTC* if, for any language $L \subseteq \Sigma^*$ satisfying $FL = LF$, there exists a language $K \subseteq \Sigma^+$ and sets I, J of nonnegative integers such that

$$F = \bigcup_{i \in I} K^i, \qquad L = \bigcup_{j \in J} K^j.$$

(The terminology refers to [1], where the commutation of two polynomials over noncommuting variables is investigated.) It is shown in [4] that every three-word code possesses BTC. We conclude with the following open problems from [4].

Open problem. Does every code possess BTC?

Open problem. Does every three-word language possess BTC?

Acknowledgements We are obliged to the referee for the careful reading of the paper and many valuable suggestions. Discussions with Cezar Campeanu are gratefully acknowledged.

References

[1] Bergman, G.; Centralizers in free associative algebras, *Trans. Amer. Math. Soc.* 137 (1969) 327-344.

[2] Bucher, W., Maurer, H. A., Culik, K., II and Wotschke, D.; Concise description of finite languages, *Theoret. Comput. Sci.* 14 (1981), no. 3, 227–246.

[3] Choffrut, C., Karhumäki, J. and Ollinger, N.; The commutation of finite sets: a challenging problem, TUCS Technical Report 303 (1999), to appear in *Theoret. Comput. Sci.*

[4] Karhumäki, J. and Petre, I.; On the centralizer of a finite set, Springer *LNCS* 1853 (2000) 536-546.

[5] Mateescu, A., Salomaa, A. and Yu, S.; On the decomposition of finite languages, TUCS Technical Report 222 (1998).

[6] Rozenberg, G. and Salomaa, A.; (eds) *Handbook of Formal Languages*, Springer, Berlin, New York, 1997.

[7] Salomaa, A.; *Theory of Automata*, International Series of Monographs in Pure and Applied Mathematics, Vol. 100 Pergamon Press, Oxford-New York-Toronto, 1969.

[8] Salomaa, A.; *Formal Languages*, Academic Press, New York, London, 1973.

[9] Salomaa, A. and Yu, S.; On the decomposition of finite languages, DLT 99 Preproceedings, Aachener Informatik-Berichte 99-5 (1999) 8-20. Appears also in: Rozenberg, G. and Thomas, W.; (eds.) *Developments in Language Theory*, World Scientific, 2000, 22-31.

[10] Shyr, H.J.; *Free Monoids and Languages*, Hon Min Book Company, Taichung, Taiwan R.O.C., 1991.

Received October, 2000