The Expressibility of Languages and Relations by Word Equations

JUHANI KARHUMÄKI

Turku University, Turku, Finland

FILIPPO MIGNOSI

Universitá di Palermo via Archirafi, Palermo, Italy

AND

WOJCIECH PLANDOWSKI

Turku University, Turku, Finland

Abstract. Classically, several properties and relations of words, such as "being a power of the same word", can be expressed by using word equations. This paper is devoted to a general study of the expressive power of word equations. As main results we prove theorems which allow us to show that certain properties of words are not expressible as components of solutions of word equations. In particular, "the primitiveness" and "the equal length" are such properties, as well as being "any word over a proper subalphabet".

Categories and Subject Descriptors: F.4.m **[Mathematical Logic and Formal Languages]:** Miscellaneous

General Terms: Theory

Additional Key Words and Phrases: Word equations

1. *Introduction*

Several authors in the existing literature, cf. Lothaire [1983], used word equations in order to describe properties and relations of words, but, to our

A preliminary version appeared in *Proceedings of ICALP'97*. Lecture Notes in Computer Science, vol. 1256. Springer-Verlag, New York, pp. 98–109.

The work of J. Karhumäki and F. Mignosi was supported by Academy of Finland under grant 14047.

The work of W. Plandowski was supported by the grant KBN 8T11C03915. This research was done while W. Plandowski from Instytut Informatyki, Banacha 2, 02-097 Warszawa, Poland, visited as a post doc researcher at Turku Centre for Computer Science (TUCS).

Authors' addresses: J. Karhumäki and W. Plandowski, Turku Centre for Computer Science and Department of Mathematics, Turku University, 20 014, Turku, Finland, e-mail: karhumak@cs.utu.fl; wojtekpl@mimuw.edu.pl; F. Mignosi, Dipartimento di Matematica el Applicazioni, Universitá di Palermo via Archirafi, 90 123 Palermo, Italy, e-mail: mignosi@altair.math.unipaiit.

knowledge, no attempt to synthesis or of a systematization of this topic has been done. This was emphasized also in a recent survey [Choffrut and Karhumäki 1997], where some results of the field were collected.

Classical relations on words that are characterized as solution sets of word equations are for instance, "two words X and Y are powers of the same word" if and only if they constitute a solution of the equation $XY = YX$, and "two words X and Y are conjugates" if and only if they constitute a solution of the equation $XZ = ZY$. In the first case, we need no extra variables, while in the second case an additional variable is needed, see Example 4. As above, we identify names of variables and particular solutions of an equation.

Motivated by above, we say that a property of words—either a language $L \subseteq \Sigma^*$ or a k-ary relation $\mathcal{R} \subseteq (\Sigma^*)^k$—is *expressible* by a word equation, if there exists an equation e with $t \geq k$ variables over Σ such that:

—L coincides with the values of a fixed component of all solutions of e, or

—$\mathcal{R}$ coincides with the values of k fixed components of all solutions of e.

Obviously, languages are k-ary relations with $k = 1$, but, due to the importance of this particular case, we have chosen to define those separately. We allow e to contain constants from Σ. An important feature here is also that t can be larger than k, that is, additional variables are allowed. This increases essentially the expressive power of equations, and in particular makes it much easier to express certain properties by equations.

As an illustration we recall the following: The union of solution sets of two equations can be expressed as a solution set of one equation, as was shown in Büchi and Senger [1986/1987] using 4 additional variables, and later improved to require only 2 additional ones by S. Grigorieff (personal communications), cf. also Choffrut and Karhumäki. Similarly, the inequality, that is the set of t-tuples of words which does not satisfy a given equation e with t variables, can be expressed as a union of the solution sets of finitely many equations, each of them using 3 extra variables, cf. for example, Choffrut and Karhumäki [1997], and consequently the inequality is expressible by one equation if additional variables are allowed.

This way of expressing relations on words using word equations is very natural and resembles the way of expressing enumerable relations on integers by diophantine equations. However, the expressive power of our method is weaker. Namely, while diophantine equations can express all recursively enumerable sets (of integers), cf. Matijasevich [1970/1971], the word equations can express only recursive relations on words due to Makanin's result, cf. Makanin [1977]. And actually our results show that not even all of those can be expressed.

A central problem in the study of the expressive power of word equations is to show that some relations are not expressible. A similar situation—a need to show that certain languages are not generated by a certain type of devices—was encountered at the early stages of the formal language theory. By now, there are a lot of tools for the latter problem cf. Harrison [1978], while there seems to be none for the former.

As the main contribution of this paper, we introduce such tools for word equations. More precisely, we prove theorems resembling pumping lemmas of formal languages, which allow to prove the nonexpressibility. Very intuitively, we show that if a given equation defines a certain language, or, in fact, just a certain word of it via a variable X, then X actually contains some "unfixed parts" which

can be filled arbitrarily, and thus leads outside the considered language. A crucial point here is to consider particular factorizations of words. For this purpose we formulate a general notion of an $\mathcal{F}$*-factorization*.

The contents of this paper are summarized as follows. In the next section, we state several properties of words that are expressible by equations, including some closure properties of expressible languages, such as the closure under catenation, union and Kleene star of a word. Most of the material in this section can be considered as a folklore, although we have at least one new proof.

Section 4 is devoted to introduce our notion of $\mathcal{F}$-factorization. Then, in Section 5, we prove our main results, namely tools for showing the nonexpressibility. In Section 6, we use our theorems to show that particular languages or relations, such as "the set of primitive words", "the language $(a \cup b)^*$ over $\{a, b, c\}$" or "the relation equal length", are not expressible. As a consequence we conclude that expressible languages are not closed under operations of Kleene star, complementation or shuffle. Next, we compare the family of expressible languages to a few much studied families, and finally, in Section 7, we state several open problems.

This paper assumes that the reader is familiar with basics both on combinatorics of words and on formal language theory. For the former, we refer to Choffrut and Karhumäki [1997] and Lothaire [1983], and for the latter to Harrison [1978] as a general reference and Rozenberg and Salomaa [1980] as a reference on D0L languages.

2. *On the Power of Expressibility*

In this section, we give several examples of properties of words that are expressible as solutions of word equations and some closure properties of languages and relations. All results presented here are either very simple or presented before, however, some of those seem to be not very generally known, and moreover we have a simplified proof.

Let Σ be an alphabet of constants and Θ be an alphabet of variables. We assume that these alphabets are disjoint. We use the convention that lower case letters represent constants and capital letters represent variables.

A word equation is a pair of words $(u, v) \in (\Sigma \cup \Theta)^* \times (\Sigma \cup \Theta)^*$ usually denoted by $u = v$. The *size* of an equation is the sum of lengths of u and v. A *solution* of a word equation $u = v$ is a morphism $h\colon (\Sigma \cup \Theta)^* \to \Sigma^*$ such that $h(a) = a$, for $a \in \Sigma$, and $h(u) = h(v)$. We say that a language L is *expressible*, if there is an equation e and a variable X such that

$$L = \{h(X) : h \text{ is a solution of } e\}.$$

Similarly, we say that a relation $\mathcal{R} \in (\Sigma^*)^k$ is *expressible* by an equation e if there are variables $X_1, \ldots, X_k$ such that

$$\mathcal{R} = \{(h(X_1), \ldots, h(X_k)) : h \text{ is a solution of } e\}.$$

The notion of the expressibility depends on the sizes of the alphabets Σ and Θ. In this paper we concentrate on the case when the alphabet Σ is finite. We also assume that $|\Sigma| \geq 2$. In the case of a unary alphabet, all expressible languages are trivially regular. The family of expressible languages over the alphabet Σ is denoted by $\mathcal{L}(\Sigma)$.

Example 1. The properties

—W contains a square, that is, W is not square-free, and

—those words W in $\{a, b, c\}^*$, which contain the letter c,

are expressible. Indeed, the former is obtained from the equation $W = XUUY$ under the extra condition $U \neq \epsilon$, where ϵ stands for the empty word, so that, by Theorem 6, the whole property can be encoded into one equation. The latter one is expressed by the equation $W = XcY$.

Example 2. Every finite and co-finite language over a finite alphabet Σ is expressible. Indeed, for $L = \{w_1, \ldots, w_t\} \subseteq \Sigma^*$, the languages L and $\Sigma^* - L$ are expressed respectively by the formulas

$$\bigvee_{i=1}^{t} X = w_i$$

and

$$\left(\bigvee_{w \neq w_i, |w| \leq N} X = w \right) \text{ or } \left(\bigvee_{|w|=N+1} X = wY \right),$$

where $N = \max\{|w_i| : i = 1, \ldots, t\}$. As above, Theorem 6 makes it possible to express these formulas using only one equation.

Example 3. The properties

—W is imprimitive, and

—W is not minimal in its conjugacy class with respect to the lexicographic ordering $\prec$

are expressible, too. This follows, as above, from the formula

$$WZ = ZW \text{ and } W = ZZT \text{ and } Z \neq \epsilon,$$

and

$$W = UV \text{ and } W' = VU \text{ and } W' \prec W,$$

after the observation that the relation $W' \prec W$ is expressible by the formula

$$\bigvee_{a \prec b} (W' = RaT \text{ and } W = RbT').$$

Example 4. The relation "X and Y are conjugates" is expressible by the equation $XZ = ZY$, but it is not expressible by any equation containing only two variables, that is, an additional variable Z is needed. The proof is by a contradiction. Suppose that it is expressible by an equation e containing two variables X and Y. Put $X = ab$ in the equation yielding an equation with one variable Y. The only solutions of this equation are ba and ab, but this is impossible since either this equation is an identity and thus all words are solutions to it, or after deleting a common prefix of both sides of the equation it is in the form $wY \cdots = Y \cdots$ with $w \neq \epsilon$ being a constant word. Then any

nonempty solution of it starts with the first letter of w. This contradicts the assumption that ab and ba are solutions of it.

After these examples, we formulate several closure properties of expressible languages and relations. Our first result is very easy.

THEOREM 5. *The family of expressible languages is closed under the following operations*: *catenation*, *cyclic closure*, *Kleene star of a single word*, *and taking the reverses of words in the language*.

PROOF. The catenation of two expressible languages is obtained by renaming the variables of one of the equations and adding a new equation $X = YZ$, where X is a new variable and variables Y and Z give the original languages.

The cyclic closure of L, that is, $cycle(L) = \{w : w = uv \text{ with } vu \in L\}$ is obtained by adding the equations $X = YZ$, $X' = ZY$ and changing the expressing variable from X to X'.

Finally, the Kleene star of a word $w = p^t$, for a primitive word p, is expressed by the formula $wX = Xw$ and $X = Y^t$, and the reverse of a language can be produced by the equation in which both sides are the reverses of the sides of the original equation. □

In order to state a few more closure properties, we extend the notion of an equations to that of Boolean formula of equations. We say that ϕ is a *Boolean formula of equations* with range and variable alphabets Σ and Θ, respectively, if it is obtained from equations over alphabets Σ and Θ by the operations of disjunction, conjunction and negation. A morphism $h: (\Sigma \cup \Theta)^* \to \Sigma^*$ such that $h(a) = a$ for each $a \in \Sigma$ is a *solution* of ϕ if and only if ϕ gets the value true whenever the values of the equations of ϕ get values true or false depending on whether or not h is a solution of the corresponding equation. Now, a relation $\rho \subseteq \Sigma^k$ is *expressible* by a Boolean formula ϕ iff there are variables $X_1, \ldots, X_k$ such that

$$\rho = \{(h(X_1), \ldots, h(X_k)) : h \text{ is a solution of } \phi\}.$$

The definition of expressibility by a Boolean formula is analogous to the definition of expressibility by one equation. In particular the formula ϕ may contain other variables than $X_1, \ldots, X_k$.

Our next result, which we have already used several times is an important tool to show relations to be expressible.

THEOREM 6. *Let* $e : u = v$ *and* $e' : u' = v'$ *be two equations*. *Then*

(*i*) *A relation expressible by* e *and* e' *is expressible by a single equation containing only variables of* e *and* e'.

(*ii*) *A relation expressible by* e *or* e' *is expressible by a single equation containing variables of* e *and* e' *and two additional ones that do not occur in* e *or* e'.

(*iii*) *The relation satisfying* $u \neq v$ (*not* $u = v$) *is expressible by a single equation using variables of* $u \neq v$ *and a finite number of variables that do not occur in* $u \neq v$.

PROOF

(i) This follows from a well-known construction, cf. for example, Khmelevski [1971], using a *pairing function*, $\langle , \rangle$ defined by

$$\langle \alpha, \beta \rangle = \alpha a \beta \alpha b \beta \text{ with } a, b \in \Sigma, a \neq b.$$

Then, as is easy to see, we have

$$u = v \text{ and } u' = v' \Leftrightarrow \langle u, v \rangle = \langle u', v' \rangle,$$

proving part (i).

(ii) First observe that, due to the equivalence

$$u = v \text{ or } u' = v' \Leftrightarrow uv' = vv' \text{ or } vu' = vv', \tag{1}$$

we may assume that the right hand sides of the considered equations are the same, that is, $v = v'$. Now we restrict the function $\langle\ \rangle$ into one variable by setting

$$\langle \alpha \rangle = \alpha a \alpha b \text{ with } a, b \in \Sigma, a \neq b.$$

Then, for each α, the shortest period, cf. Choffrut and Karhumäki [1997], of $\langle \alpha \rangle$ is longer than half of its length, in particular $\langle \alpha \rangle$ is primitive. Now, the result is a consequence of the following equivalence:

$$u = v \text{ or } u' = v \Leftrightarrow \exists Z, Z' : X = ZYZ', \tag{2}$$

where

$$Y = \langle uu' \rangle^2 v \langle uu' \rangle^2$$

and

$$X = \langle uu' \rangle^2 u \langle uu' \rangle^2 u' \langle uu' \rangle^2.$$

The proof of (2) follows directly from the facts that the word $\langle uu' \rangle^2$ is a prefix and a suffix of Y and that it occurs in X in exactly three places: as a prefix, in the middle, and as a suffix. The last fact is based on two properties of the word $\langle uu' \rangle$. First, since it is primitive it occurs inside the word $\langle uu' \rangle^2$ in exactly two places: as a prefix and as a suffix. Second, the word $\langle uu' \rangle^2$ cannot occur in $\langle uu' \rangle\ u\ \langle uu' \rangle$ or in $\langle uu' \rangle\ u'\ \langle uu' \rangle$ since $\langle uu' \rangle$ is at least twice longer than both u and u', and the shortest period of $\langle uu' \rangle$ is longer than half of its length.

(iii) This again is based on a well-known fact, cf. for example, Culik and Karhumäki [1983],

$$\begin{aligned} u \neq v \quad \Leftrightarrow \quad & \exists Z, Z', Z'': \\ & \left(\bigvee_{a \in \Sigma} u = vaZ \right) \text{ or} \\ & \left(\bigvee_{a \in \Sigma} v = uaZ \right) \text{ or} \\ & \left(\bigvee_{a,b \in \Sigma, a \neq b} (u = ZaZ' \text{ and } v = ZbZ'') \right), \end{aligned}$$

and parts (i) and (ii). □

Theorem 6 deserves a few comments. First, we have a new proof of case (ii), which is a simplification of the proof presented in Choffrut and Karhumäki [1997], which, in turn, was based on ideas of S. Grigorieff (personal communication). Second, it gives more closure properties of expressible languages and relations, such as

THEOREM 7. *Any language or relation of words expressible by a Boolean formula of equations is expressible by a single equation.*

PROOF. Follows directly from Theorem 6. Indeed, we first replace a given Boolean formula ϕ by an equivalent formula ϕ' in disjunctive (or conjunctive) normal form, that is by an equivalent formula where negations occur only on the lowest level. This means that ϕ' is built from equations and inequalities by operations of disjunctions and conjunctions. Now, Theorem 6 becomes immediately applicable to built a single equation with extra variables expressing the same relation as ϕ. □

Two more concrete closure properties of the family of expressible languages are immediate consequences of cases (i) and (ii) of Theorem 6.

THEOREM 8. *The expressible languages are closed under finite intersections, and finite unions.*

With the application of case (iii) of Theorem 6 in proving closure properties of expressible languages one has to be careful. The consequence of this case is that the complement of the relation defined by an equation $u = v$ using *all variables of the equation* is expressible by a single equation (using additional variables). This, however, does not mean that expressible languages are closed under the complementation. Consider, for instance, a binary relation ρ expressible by an equation e containing only two variables. Clearly, the language $L = \{x|(x, y) \in \rho\}$ is expressible by e. By (iii) of Theorem 6, the complement of ρ and the language $L' = \{z|(z, h) \notin \rho\}$ are expressible, but L' is not necessarily the complement of language L (L and L' could even coincide).

We shall show in Section 6 that expressible languages are not closed under complementation. Of course, in some special cases, such a closure might hold.

The fact that expressible relation are not closed under complementation unless relation is defined by equation using no extra variables, is analogous to the well-known fact that recursively enumerable relations are not closed under complementation while recursive ones are. Indeed, recursively enumerable relations are obtained from recursive ones allowing to use extra components and existential quantifications.

3. *Expressibility of Languages by Equations with Two Variables*

In this section, we introduce technical tools and apply those to languages expressible using only two variables. We consider a fixed equation $u = v$ and fix the lengths of the components of its solution h by a vector $\vec{z}$ of nonnegative integers. This fixes also the length of each word of the form $h(w)$, in particular the lengths of both sides of $h(u) = h(v)$. Next we use the identity $h(u) = h(v)$ in Σ^*, that is identify the corresponding letters on both sides of this identity, to define an equivalence relation $\mathcal{R}_{\vec{z}}$ on positions of $h(u)$. The positions in the

equivalence classes are to be filled by the same constant. The constant is uniquely determined if one of the positions in the class corresponds to a constant in an original equation. Otherwise, the constant can be chosen arbitrarily. Moreover, the positions in such a class can be filled by any word.

We formalize the above as follows: Assume that an equation e contains t variables $X_1, X_2, \ldots, X_t$ and $\vec{z} = (z_1, \ldots, z_t)$ is a vector of t natural numbers. We say that h is a $\vec{z}$-*solution* of e if h is a solution of e and $|h(X_j)| = z_j$, for $1 \leq j \leq t$. For a vector $\vec{z} = (z_1, \ldots, z_t)$, we define a function $|\cdot|_{\vec{z}}: (\Theta \cup \Sigma)^* \to N$ by

$$|u|_{\vec{z}} = \begin{cases} z_m & \text{if } u = X_m \in \Theta, \\ 1 & \text{if } u \in \Sigma, \\ \sum_{k=1}^{s} |u_k|_{\vec{z}} & \text{if } u = u_1 u_2 \cdots u_s \text{ with } u_j \in \Theta \cup \Sigma. \end{cases}$$

In other words $|w|_{\vec{z}}$ is the length of the word $h(w)$ when h is a $\vec{z}$-solution and $w \in (\Theta \cup \Sigma)^*$.

Now, assume that we are given an equation $u_1 \cdots u_k = v_1 \cdots v_s$ over t variables and a vector $\vec{z} \in N^t$ such that $|u|_{\vec{z}} = |v|_{\vec{z}}$. We define a function $left_{\vec{z}}$: $\{1, \ldots, |u|_{\vec{z}}\} \to N \times (\Theta \cup \Sigma)$ in the following way:

$$left_{\vec{z}}(j) = (r, x) \text{ iff}$$
$$|u_1 \cdots u_p|_{\vec{z}} < j \leq |u_1 \cdots u_{p+1}|_{\vec{z}} \text{ and } r = j - |u_1 \cdots u_p|_{\vec{z}} \text{ and } u_{p+1} = x.$$

Similarly, we define the function $right_{\vec{z}}$:

$$right_{\vec{z}}(j) = (r, x) \text{ iff}$$
$$|v_1 \cdots v_p|_{\vec{z}} < j \leq |v_1 \cdots v_{p+1}|_{\vec{z}} \text{ and } r = j - |v_1 \cdots v_p|_{\vec{z}} \text{ and } v_{p+1} = x.$$

Finally, an equivalence relation $\mathcal{R}_{\vec{z}}$ on positions $\{1 \cdots |u|_{\vec{z}}\}$ is the transitive and symmetric closure of the relation $\mathcal{R}'_{\vec{z}}$ defined by

$$i \mathcal{R}'_{\vec{z}} j \text{ iff } left_{\vec{z}}(i) = right_{\vec{z}}(j) \text{ or } left_{\vec{z}}(i) = left_{\vec{z}}(j) \text{ or } right_{\vec{z}}(i) = right_{\vec{z}}(j).$$

We say that a position i *belongs* to a variable X if either $left_{\vec{z}}(i) = (j, X)$ or $right_{\vec{z}}(i) = (j, X)$, for some j. Let $\mathcal{X}$ be an equivalence class of the relation $\mathcal{R}_{\vec{z}}$. We say that $\mathcal{X}$ *corresponds* to a constant a if there is a position i in $\mathcal{X}$ such that either $left_{\vec{z}}(i) = (1, a)$ or $right_{\vec{z}}(i) = (1, a)$.

Example 9. Consider an equation $e: aX_1X_2bX_1 = X_3X_4X_3$. Let $\vec{z} = (2, 4, 5, 0)$. Then the values of the functions $left_{\vec{z}}$ and $right_{\vec{z}}$ are as listed below.

	1	2	3	4	5
$left_{\vec{z}}$	$(1, a)$	$(1, X_1)$	$(2, X_1)$	$(1, X_2)$	$(2, X_2)$
$right_{\vec{z}}$	$(1, X_3)$	$(2, X_3)$	$(3, X_3)$	$(4, X_3)$	$(5, X_3)$

	6	7	8	9	10
$left_{\vec{z}}$	$(3, X_2)$	$(4, X_2)$	$(1, b)$	$(1, X_1)$	$(2, X_1)$
$right_{\vec{z}}$	$(1, X_3)$	$(2, X_3)$	$(3, X_3)$	$(4, X_3)$	$(5, X_3)$

The equivalence classes of $\mathcal{R}_{\vec{z}}$ are $\mathcal{X} = \{1, 6\}$, $\mathcal{Y} = \{3, 5, 8, 10\}$ and $\mathcal{Z} = \{2, 4, 7, 9\}$. The classes $\mathcal{X}$ and $\mathcal{Y}$ correspond to the constants a and b, respectively, since $left_{\vec{z}}(1) = (1, a)$ and $left_{\vec{z}}(8) = (1, b)$, respectively. The equivalence class $\mathcal{Z}$ does not correspond to any constant. Hence, the positions in $\mathcal{Z}$ can be filled with any letter and, by case (iv) of Lemma 10, they can be replaced by any word as well. This gives the following family of solutions of the equation e:

$$X_1 = \beta a,\ X_2 = \beta b a \beta,\ X_3 = a \beta b \beta b,\ X_4 = \epsilon,$$

where β is an arbitrary word.

The above procedure, illustrated in Example 9, can be seen as a *method of filling the positions* of the variables in an equation. This simple method, which was first used in Lentin [1972], can be used, for example, to give a very illustrative proof for the periodicity theorem of Fine and Wilf, cf. for example, Choffrut and Karhumäki [1997].

Now the following lemma is obvious. Denote by $w[i]$ the ith letter of the word w.

LEMMA 10. *Let $\mathcal{C}$ be an equivalence class of the relation $\mathcal{R}_{\vec{z}}$ connected to an equation $e: u = v$. Then the following conditions are satisfied*:

(*i*) *For any two positions $i, j \in \mathcal{C}$ and a $\vec{z}$-solution h of e, $h(u)[i] = h(u)[j]$.*
(*ii*) *If $\mathcal{C}$ corresponds to a constant a and $i \in \mathcal{C}$, then for each $\vec{z}$-solution h of e, $h(u)[i] = a$.*
(*iii*) *If $\mathcal{C}$ corresponds to two different constants a and b, then the equation e has no $\vec{z}$-solution.*
(*iv*) *If $\mathcal{C}$ does not correspond to any constant and e has a $\vec{z}$-solution h, then replacing in h the letters corresponding to positions in $\mathcal{C}$ by any word yields a new solution of e.*

Note, that in case (iv) new solutions obtained need not be $\vec{z}$-solutions anymore.

In a formulation of our results, we need a notion of a pattern language from Angulin [1979], cf. also Jiang et al. [1995]. A *pattern* is a word over the alphabet $\Theta \cup \Sigma$. A *pattern language* generated by a pattern w is the set of all words that are morphic images of w under morphisms $h: (\Theta \cup \Sigma)^* \to \Sigma^*$ satisfying $h(a) = a$, for a in Σ. In particular, it is natural to denote by $p((\Sigma^*)^k)$ the pattern language generated by a pattern $p(X_1, X_2, \ldots, X_k)$ containing k variables $X_1, X_2, \ldots, X_k$. We have an obvious connection:

Example 11. Each pattern language is expressible. Let u be a pattern and Z be a variable that does not occur in u. A variable Z in equation $Z = u$ expresses the pattern language generated by u.

We also need an auxiliary lemma that follows rather straightforwardly from Lemma 10 and which holds for any number of variables.

LEMMA 12. *Let L be an expressible language via a variable X in an equation e. Suppose that there is no one-variable pattern $p(Y)$ such that $p(\Sigma^*) \subseteq L$. Then, for each vector $\vec{z}$, there is a word $w \in L$ such that, for each $\vec{z}$-solution h of e, the equation $h(X) = w$ is satisfied.*

PROOF. Consider a relation $\mathcal{R}_{\vec{z}}$ for $\vec{z}$-solutions of the equation e. Suppose now that h is a $\vec{z}$-solution of e. If there is an equivalence class of $\mathcal{R}_{\vec{z}}$ which does

not contain a position corresponding to a constant of e, then by Lemma 10 all positions in the class can be filled with the same arbitrary word, and each choice of such a word creates a new solution of e. Hence, if such an equivalence class contains positions belonging to the variable X, then there is a pattern p which can be chosen to contain just one unknown, say $p = p(Y)$, such that $p(\Sigma^*) \subseteq L$. If there is no such an equivalence class, then all positions in X are in equivalence classes corresponding to constants of e. Then, by Lemma 10, all symbols of $h(X)$ are uniquely determined, which was to be proved. □

Now denoting by $\#_L(n)$ the number of words of length n in the language L, we are ready to prove the main result of this section.

THEOREM 13. *Let L be expressible by an equation on two variables. Then either $\#_L(n) = O(n)$ or there is a pattern $p(Y)$ with one variable such that $p(\Sigma^*) \subseteq L$.*

PROOF. Let a language L be expressible via a variable X in an equation $e(X, Z)$ of length d and suppose that there is no pattern p such that $p(\Sigma^*) \subseteq L$. Let x be a word in L of length n. Consider the equation $e(x, Z)$ with one variable Z. Its size does not exceed dn, and since $x \in L$ it is solvable. It is not difficult to see, cf. Eyono Obono et al. [1994], see also Khmelevski [1971/1976], that there is a solution z of $e(x, Z)$ whose length does not exceed four times the length of the equation, that is, does not exceed $4dn$. Now the result follows from the fact that the number of pairs $(n, |z|)$, with $|z| \leq 4dn$, is linear with respect to n and, by Lemma 12, each of these corresponds to at most one word in L. □

As a straightforward consequence of Theorem 13, we obtain a gap theorem for the function $\#_L(n)$.

THEOREM 14. *Let L be expressible by an equation with two variables. Then either $\#_L(n) = O(n)$ or $\#_L(an + b) = 2^{\Omega(n)}$ for some constants a and b.*

Theorem 14 deserves a few comments. First, observe that the complexity function $\#_L(n)$ is always at most exponential, that is in the class $2^{O(n)}$. Second, it provides a tool to show that some languages are not expressible with only one additional unknown.

Example 15. The language $L = a^*b^*c^*$ is not expressible by equations containing two unknowns though it is expressible. The nonexpressibility by equations with two variables is a consequence of Theorem 14 and the fact that

$$\#_L(n) = |\{(x, y, z) : x + y + z = n \text{ and } x, y, z \geq 0\}| = \binom{n+2}{2} = \Theta(n^2).$$

On the other hand, the language L is expressible via a variable U in formula

$$U = XYZ \text{ and } aX = Xa \text{ and } bY = Yb \text{ and } cZ = Zc.$$

4. *$\mathcal{F}$-factorizations*

In this section, we introduce a general notion of an $\mathcal{F}$-factorization, which, we hope, might turn out useful in many applications. For our considerations, certain types of such factorizations are of a crucial importance.

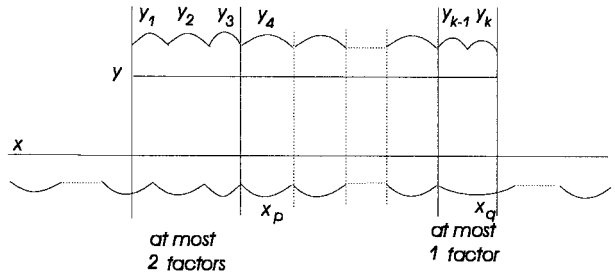

FIG. 1. Illustration of the condition (iii) with l = 3 and r = 2.

We recall that an *F-factorization* of a word w is any sequence $w_1, \ldots, w_k$ of words from a language F such that $w = w_1 \cdots w_k$. We generalize it as follows: Let $\mathcal{F}$ be a property of sequences of words. We say that a sequence $w_1, \ldots, w_k$ is an $\mathcal{F}$-factorization of w if $w = w_1 \cdots w_k$ and the sequence satisfies $\mathcal{F}$. Further we say that a property $\mathcal{F}$ defines a *synchronizing factorization*, or briefly that $\mathcal{F}$ is *synchronizing*, if the following holds:

(i) It is *complete*, that is, each word possesses an $\mathcal{F}$-factorization.
(ii) It is *uniquely deciphering*, that is, each word has at most one $\mathcal{F}$-factorization,
(iii) It satisfies the following *synchronizing condition* for some nonnegative integer parameters l and r. Let $y_1, \ldots, y_k$ and $x_1, \ldots, x_s$ be $\mathcal{F}$-factorizations of words y and x, respectively. If $k > l + r$ and y occurs in x at position i, then there is a pair of numbers l', r' satisfying $l' \leq l$ and $r' \leq r$ such that the following conditions hold. Denote $u = y_1 \cdots y_{l'}$ and $v = y_{k-r'+1} \cdots y_k$. Then,

— positions $i + |u|$ and $i + |y| - |v|$ in x are starting positions of factors, say x_p and x_q, respectively,
— the sequences of factors $x_p, \ldots, x_{q-1}$ and $y_{l'+1}, \ldots, y_{k-r'+1}$ are identical, that is,

$$q - 1 - p = k - r' - l' \text{ and } x_{p+j} = y_{l'+1+j}, \qquad \text{for} \quad 0 \leq j \leq k - r' - l',$$

— the occurrence of u at position i in x covers at most $l - 1$ factors of x, that is, u is a suffix of $x_{\max\{p-l,1\}} \cdots x_{p-1}$,
— the occurrence of v at position $i + |y| - |v|$ in x covers at most $r - 1$ factors of x, that is, v is a prefix of $x_q \cdots x_{\min\{q+r-1,s\}}$.

The above definition requires a few comments. First, notice that a synchronizing factorization is always "synchronizing" with finite values of l and r. In case $l = 3$ and $r = 2$ the property (iii) can be illustrated as in Figure 1. Here sequences of arches represent $\mathcal{F}$-factorizations. The word $u = y_1y_2y_3$ covers at most 2 factors of x and the word $v = y_{k-1}y_k$ covers at most 1 factor of x.

Second, it would be natural to call $\mathcal{F}$-factorizations satisfying only (iii) synchronizing. However, as we shall see, we need also conditions (i) and (ii) in our considerations, although the condition (iii) is the most central one. Therefore we included all three conditions to the notion of a synchronizing factorization.

Third, our notion of an $\mathcal{F}$-factorization is connected to, but not the same as, that of a factorization of a free monoid, cf. Berstel and Perrin [1985] and

Lothaire [1983]. These factorizations are used to decompose free monoids, while in our considerations a focus is on factorizations of a single word.

Fourth, our notion of a synchronizing $\mathcal{F}$-factorization resembles factorizations of words defined by uniformly synchronous codes, cf. Berstel and Perrin [1985]. Clearly, factorizations of words defined by codes satisfy (ii), but do not have to satisfy (i). However, if we restrict our general notion of an $\mathcal{F}$-factorization to factorizations defined by codes then the ones which satisfy our synchronizing condition (iii) are precisely the factorizations which are defined by uniformly synchronous codes. Indeed, our condition (iii) can be rewritten for a code X in the following way:

— Let $y \in X^k$ and $x = x'yx'' \in X^s$. If $k > l + r$ then there is a pair of numbers l', r' satisfying $l' \leq l$, $r' \leq r$ such that the following conditions hold:
  — $y = uy'v$ for $(u, v) \in X^{l'} \times X^{r'}$, $y' \in X^*$,
  — $x'u$, $vx'' \in X^*$,
  — $x'u = x_1 \cdots x_{p-1}$ where $x_t \in X$, for $1 \leq t \leq p - 1$, and u is a suffix of $x_{\max\{p-l,1\}} \cdots x_{p-1}$.
  — $vx'' = x_q \cdots x_s$ where $x_t \in X$, for $q \leq t \leq s$, and v is a prefix of $x_q \cdots x_{q+\min\{q+r-1,s\}}$.

It is easy to see that this condition is equivalent to the following one. (Below, l and r are not necessarily the same as in the previous condition.)

—There are nonnegative integers l and r such that, for each pair $(u, v) \in X^l \times X^r$ and for a word $y' \in X^*$, if $x'uy'vx'' \in X^*$, then $x'u$, $vx'' \in X^*$.

It is not difficult to see that this condition is equivalent to the following one:

—There is a nonnegative integer s such that for each pair $(u, v) \in X^s \times X^s$, if $x'uvx'' \in X^*$, then $x'u$, $vx'' \in X^*$.

The last condition is, in turn, one of the possible definitions of uniformly synchronous codes, cf. Berstel and Perrin [1985, Proposition 2.4, p. 331].

We still need two more notions. We say that the $\mathcal{F}$-factorization $w_1, \ldots, w_k$ of w *synchronizes* with a pattern p if p is of the form $p = u_1u_2 \cdots u_k$, where each u_i is either a variable or equal to w_i, and there are no $i, j \in \{1, \ldots, k\}$ such that $w_i \neq w_j$ and $u_i = u_j$. Finally, we set $n_{\mathcal{F}}(w)$ to denote the number of *different* words in the $\mathcal{F}$-factorization of a word w, that is, $n_{\mathcal{F}}(w)$ is the cardinality of the set of words in the $\mathcal{F}$-factorization of w.

5. *Main Results*

This section is devoted to prove some pumping-like theorems of expressible languages. These are achieved by using tools of the previous sections, namely the method of filling positions of variables and synchronizing $\mathcal{F}$-factorizations. The latter involves methods to generalize techniques from Bulitko [1970], cf. also Koscielski and Pacholski [1996], which were used to prove an upper bound for the index of the periodicity of a minimal solution of a word equation.

Now we are ready to state our first tool to show the nonexpressibility.

THEOREM 16. *Let L be expressible by an equation e and let $\mathcal{F}$ be a property defining a synchronizing factorization. Then there exists a constant $k = k(\mathcal{F}, e)$ such*

that, for each $w \in L$ *with* $n_{\mathcal{F}}(w) > k$, *there is a one-variable pattern* $p(X)$ *satisfying* $w \in p(\Sigma^*) \subseteq L$.

PROOF. The proof uses similar ideas as that of Lemma 12. Let $\mathcal{F}$ be synchronizing with the parameters l and r, and let d be the size of the equation e. We shall show that the theorem holds for

$$k = (l + r - 1)d + l + r.$$

Consider any solution h of $e : u = v$, and assume that a variable X defines L. Let $f_1, f_2, \ldots, f_n$ be the $\mathcal{F}$-factorization of the word $z = h(u) = h(v)$ and $y_1, \ldots, y_t$ be the $\mathcal{F}$-factorization of the word $h(Y)$, for a variable Y. In each of the $\mathcal{F}$-factorizations of the words $h(Y) = y_1 \ldots y_t$ we call factors (if any)

$$y_i \text{ with } l + 1 \leq i \leq t - r$$

inside factors, and all the other *end* factors. Then, since $\mathcal{F}$ is synchronizing, the inside factors of each occurrence of $h(Y)$ in z corresponding to an occurrence of Y on the left or right hand sides of e, form a subsequence in the $\mathcal{F}$-factorization of z.

Next, as in Section 2, we introduce two partial functions *left* and *right*. They are defined on positions of factors in the $\mathcal{F}$-factorization of z, that is, on the set $1, \ldots, n$, and not on positions of letters in z as they were defined in Section 2. We set, for $i = 1, \ldots, n$, $left(i) = (j, Y)$ if the factor f_i matches the inside factor y_j of $h(Y)$ for some occurrence of Y on the left hand side of e (i.e., $f_i = y_j$), and $left(i)$ is undefined if f_i does not correspond to an inside factor of any word $h(Y)$ on the left hand side of e. Similarly, $right(i) = (j, Y)$ if the factor f_i corresponds to an inside factor y_j of $h(Y)$ of some occurrence of Y from the right-hand side of e, and $right(i)$ is undefined if f_i does not correspond to an inside factor of any word $h(Y)$ on the right-hand side of e.

Now let us call a factor f_i in the $\mathcal{F}$-factorization of z *proper* if both $left(i)$ and $right(i)$ are defined. This implies that no proper factor overlaps with any of the inside factors. *Unproper* factors either contain some constant from either of the sides of e, or they overlap some end factor of some variable. By the synchronizing condition of $\mathcal{F}$, there are at most $(l + r - 1)d$ unproper factors of z.

We continue by defining an equivalence relation $\mathcal{R}$ on the set $\{1, \ldots, n\}$. This is done as it was done for $\mathcal{R}_{\bar{z}}$ in Section 2. The only thing to be noticed is that our partial functions $left(i)$ and $right(i)$ must be extended to total ones by setting their undefined values to $(1, f_i)$. Such an f_i is an unproper factor.

Consider now the equivalence classes of the relation $\mathcal{R}$. Recalling that X was the variable expressing L, we have two possibilities:

(i) Each equivalence class of $\mathcal{R}$ containing a position of an inside factor of $h(X)$ contains also a position of an unproper factor.
(ii) Some equivalence classes of $\mathcal{R}$ containing positions of inside factors of $h(X)$ contain only positions of proper factors.

In the first case, the number of different inside factors of $h(X)$ does not exceed the number of unproper factors, that is, does not exceed $(l + r - 1)d$. Hence, $n_{\mathcal{F}}(h(X)) \leq (l + r - 1)d + l + r = k$.

In the second case, we proceed as follows: We define from the factorization $h(X) = x_1 \cdots x_t$ a pattern p by replacing factors belonging to an equivalence class of $\mathcal{R}$ containing only positions of proper factors by a new variable Z. If there are $s > 1$ such equivalence classes, then the pattern p contains s variables, and moreover $p((\Sigma^*)^s) \subseteq L$. We have $s \geq n_{\mathcal{F}}(h(X)) - k$. Clearly, we are not obliged to treat all s equivalence classes in this way. Therefore, the above pattern p can contain less than s variables, in particular only one as was to be shown. □

Theorem 16 deserves a few comments. First, the constant in the theorem can be chosen to be

$$k = (l + r - 1)d + l + r,$$

where l and r are the parameters in the definition of the synchronizing factorization, and d is the size of the equation.

Second, the pattern p constructed in the proof clearly synchronizes with the $\mathcal{F}$-factorization in the sense defined at the end of Section 4, and moreover contains at least $n_{\mathcal{F}}(h(X)) - k$ variables.

Third, the proof uses a simple and natural idea of filling the positions of variables, and due to the notion of the synchronizing factorization this method becomes very powerful. Moreover, the fact that the $\mathcal{F}$-factorization contains many enough *different* words as factors plays an important role in the proof.

Finally, the usefulness of Theorem 16 is pointed out in the next section.

Based on the above remarks, Theorem 16 can be strengthened as shown here. As we shall see in Section 6, this stronger formulation is important in proofs of the nonexpressibility of some languages.

THEOREM 17. *Let L be expressible by an equation of length d and let $\mathcal{F}$ be a property defining a synchronizing factorization with parameters l and r. Then, for each $w \in L$ satisfying $n_{\mathcal{F}}(w) > k = (l + r - 1)d + l + r$, there exists a pattern $p(X_1, \ldots, X_s)$ with $s = n_{\mathcal{F}}(w) - k$ variables such that it synchronizes with the $\mathcal{F}$-factorization of w and satisfies $p((\Sigma^*)^s) \subseteq L$.*

Now, we move to a proof of our second main theorem. It is based on Theorem 17 and additional considerations, similar to those in Bulitko [1970] and Koscielski and Pacholski [1996] dealing with a particular $\mathcal{F}$-factorization.

Let Q be a primitive word. Then it is known [Koscielski and Pacholski 1996] that each word $w \in \Sigma^*$ can be uniquely written in the form

$$w = w_0 Q^{x_1} w_1 \cdots Q^{x_k} w_k, \tag{3}$$

where, for all i,

—w_i does not contain Q^2 as a subword,
—Q is a proper prefix of w_i if $i \neq 0$,
—Q is a proper suffix of w_i if $i \neq k$,
—$x_i \geq 0$.

This leads to at least two related ways of defining an $\mathcal{F}$-factorization. The *$\mathcal{F}_Q$-factorization* of w having the presentation (3) is defined to be

$$w_0, Q^{x_1}, \ldots, Q^{x_k}, w_k.$$

In particular, for $k = 0$, it contains only one factor, namely the word itself. The *$\mathcal{F}'_Q$-factorization* differs from the $\mathcal{F}_Q$-factorization only in the sense that the factors Q^{x_i} are replaced by products of Q's.

It is obvious that both of these $\mathcal{F}$-factorizations are synchronizing. Indeed they are so with the parameters $l = 2$ and $r = 2$. Although they look very similar, they behave quite differently from the point of view of our considerations: the $\mathcal{F}_Q$-factorization is suitable for our purposes while the $\mathcal{F}'_Q$-factorization is not.

In order to formulate our second tool to prove the nonexpressibility, we denote

$$T_Q(w) = \{x : Q^x \text{ is an element in the } \mathcal{F}_Q\text{-factorization of } w\},$$

and define the *exponent* of w with respect to Q by the formula

$$exp_Q(w) = \max(\{x : x \in T_Q(w)\} \cup \{0\}).$$

The zero is added to make the function $exp_Q(w)$ defined in case the set $T_Q(w)$ is empty.

Theorem 17 applied to the $\mathcal{F}_Q$-factorization yields easily the following result:

THEOREM 18. *Let L be expressible by an equation e and let Q be a primitive word. Then there exists a constant $k = k(e)$ such that whenever $w \in L$ and $|T_Q(w)| > k$, there exists a word u in L such that $|T_Q(u)| \leq k$, and u is obtained from w by removing from its $\mathcal{F}$-factorization some factors of the form Q^x.*

PROOF. The $\mathcal{F}_Q$-factorization is synchronizing with the values $l = 2$ and $r = 2$. Consequently, in Theorem 17, we can choose $k(e, \mathcal{F}_Q) = (l + r - 1)d + l + r = 3d + 4$, where d is the length of the equation. Therefore, if we set $k = 3d + 4$, Theorem 18 follows from Theorem 17: the word u is obtained from w by replacing in the pattern p the variables corresponding to factors of the form Q^x by the empty word and all other variables by appropriate factors from the $\mathcal{F}$-factorization of w. □

Our next goal is to prove a theorem that resembles Theorem 18, but is actually much stronger. An essential tool in that proof is to employ the ideas introduced in Bulitko [1970] and later used in Koscielski and Pacholski [1996].

THEOREM 19. *Let L be expressible by an equation e and Q be a primitive word. Then there exists a natural number $k = k(e)$ such that, for each word w in L satisfying $exp_Q(w) > k$ there is a word u in L with $exp_Q(u) \leq k$ and obtained from w by removing some occurrences of Q.*

Before presenting the proof of Theorem 19, we fix some terminology and prove an auxiliary lemma that is crucial for the proof. By a system of linear equations, we mean a system of the form

$$S : \sum_{i=1}^{t} a_{ij} x_i = b_j,$$

where $j = 1, \ldots, q$ for some $q \geq 1$, and each a_{ij} and b_j is an integer. The set $\Theta = \{x_i | i = 1, \ldots, t\}$ is the set of variables of S. A *solution* of S is any t-tuple

of *nonnegative* integers satisfying all equations of S. For a subset Θ' of Θ a vector s' of dimension $|\Theta'|$ is a *partial Θ'-solution*, or a *partial solution* for short, if there exists a solution s such that its restriction to Θ' coincides with s'. A solution s of S is called *minimal* if there does not exist any solution s' that would satisfy $s' < s$ in the natural componentwise ordering. Further we call the numbers $o(S) = t$, $s(S) = q$ and $d(S) = \max\{|b_j|, |a_{ij}|\}$ the *order*, the *size* and the *depth* of S, respectively.

Finally, we recall a formulation of the fundamental König Infinity Lemma, cf. Harrison [1978].

LEMMA 20 [THE KÖNIG INFINITY LEMMA]. *Each set of pairwise incomparable elements of N^k is finite.*

With the above terminology, we now prove our crucial lemma. In the formulation of the lemma, we interpret a set of integers as a vector in an appropriate way.

LEMMA 21. *Let Q be a primitive word and L be expressible by an equation e of size d. Then, for each $w \in L$, there exists a system of equations S_w of order at most $d + |T_Q(w)|$, of size at most $2d + |T_Q(w)|$ and of depth at most $3d$ such that*

(i) *$T_Q(w)$ forms a partial solution of S_w, and*
(ii) *for each solution s of S_w there exists a word w' in L such that $T_Q(w')$ coincides with a partial solution defined by s.*

PROOF. Let e be the equation $u = v$ and let h be a solution of e. Further, let the $\mathcal{F}_Q$-factorization of $h(u)$ and $h(X)$, for a variable X, be

$$h(u) = w_0 \cdot Q^{w_1'} \cdot w_1 \cdot Q^{w_2'} \cdots w_{t-1} \cdot Q^{w_t'} \cdot w_t, \tag{4}$$

and

$$h(X) = x_0 \cdot Q^{x_1'} \cdot x_1 \cdot Q^{x_2'} \cdots x_{k-1} \cdot Q^{x_k'} \cdot x_k, \tag{5}$$

respectively. Consequently, $T_Q(h(X)) = \{x_1', \ldots, x_k'\}$. We call all factors $Q^{x_1'}$ or $Q^{x_k'}$ in (5) *end factors* and all the other factors in (5) *inside factors*.

Next we analyze how, for a fixed i, the occurrence of the word

$$Q_i = Q \cdot Q^{w_i'} \cdot Q$$

is related to the $\mathcal{F}_Q$-factorizations (5) of the words $h(X)$, for variables X occurring in u. There are three possibilities: either Q_i is a factor of $h(X)$, or Q_i goes over at least one of the ends of $h(X)$, or it consists of constants of u.

In the first case, due to the primitiveness of Q, necessarily we have, for some j, the equality

$$w_i' = x_j'. \tag{6}$$

In the other two cases, there exists a unique minimal subword of u, say

$$\alpha_0 X_1 \alpha_1 \cdots \alpha_{m-1} X_m \alpha_m \qquad \text{with} \qquad m \geq 0,$$

where α_i's are (possibly empty) constants and X_j's are (not necessarily distinct) variables, such that Q_i is a subword of $\alpha_0 h(X_1) \alpha_1 \cdots \alpha_{m-1} h(X_m) \alpha_m$. Now, for

each $l = 1, \ldots, m$, either $T_Q(h(X_l)) = \emptyset$ or in the $\mathcal{F}_Q$-factorization of $h(X_l)$ the constant k equals 1, that is, we can write

$$h(X_l) = x_{0,l} \cdot Q^{x_l'} \cdot x_{1,l}. \tag{7}$$

Therefore, we obtain the identity

$$w_i' = \sum_{l=1}^{m} (x_l' + 2) + c, \tag{8}$$

where c is a constant not larger than d and $m \leq |u|$. Indeed, each letter of α_l, as well as each X_l with $T_Q(h(X_l)) = \emptyset$, can contribute at most one Q to Q_i. Note here that the same variable can occur in several places in $\alpha_0 h(X_1)\alpha_1 \cdots \alpha_{m-1} h(X_m)\alpha_m$ so that the same exponent x_l' may appear several times in (8).

Next, we apply the above procedure to the right-hand side of the equation, that is, with respect to v. This yields another formula for w_i', say in terms of variables y_l'. Identifying these two values of w_i' we obtain an equation of the form

$$\sum_{l=1}^{N} a_l z_l' = c',$$

where z_l''s are variables corresponding to exponents x_l' and y_l' in (7), for some variables X_l and Y_l, and $\Sigma_{l=1}^{N} |a_l| \leq |u| + |v| = d$. Moreover, the absolute value of c' is at most $3d$.

Now we carry out the above procedure for all words w_i'. This results a system of equations

$$S: \sum_{l=1}^{N_i} \alpha_{il} z_l' = c_i', \tag{9}$$

where $c_l' \leq 3d$ and $\Sigma_{l=1}^{N_i} |\alpha_{il}| \leq d$. In this system the variables are exponents from the representations (5), each of them interpreted as one variable. Some of the equations of S are, due to (6), of the form $x_i' - y_j' = 0$, and these are called *simple*. An important property of S is that the number of its nonsimple equations is, at most, the number of borders between constants and/or variables, which is at most d. This indeed follows from the construction of (8).

Next we note that the system S satisfies the properties (i) and (ii) in the lemma. Indeed, the system S was constructed under the assumption that h is a solution of e, and then the equations were obtained by analyzing how the words $Q^{w_i'}$ can match with the $\mathcal{F}_Q$-factorizations of the words $h(X)$ for all variables X. Consequently, if X' is the variable defining the language L the set $T_Q(h(X'))$ is a partial solution of S, fixing the values of those variables in S which came from the $\mathcal{F}_Q$-factorization (5) for X'. Conversely, from a solution s of S, by our construction, we obtain a solution h' of e, such that s equals to a vector obtained from $T_Q(h(X'))$ and from all exponents occurring in representations of $h(X)$ in (5). In particular, $T_Q(h(X'))$ is a partial solution.

It remains to evaluate the sizes of the parameters of the system S. Actually, before doing that we still reduce S as follows. We recall that X' is the variable expressing L and carry out the following two reduction steps:

(I) For each z'_l which comes from the representation of $h(X')$, we merge y'_i to z'_l if S contains a chain of simple equations such that the first one contains z'_l, the last one contains y'_i, and two consecutive ones have a common variable.
(II) All variables of S that do not occur in an equation containing a variable which corresponds to an exponent in (5) of $h(X')$ are deleted.

Let S_w be the system thus obtained. It follows directly that S_w still satisfies the requirements (i) and (ii). However, the solution of e giving w as the value of the variable X' is now changed. For example, if in step (II) a variable is deleted, then in the original h some factors of Q are replaced by the empty word.

Now we are ready to show that S_w is of the required form. First, as we noticed, the system S from (9) contains only at most d nonsimple equations, and so does S_w. Moreover, by reduction steps (I) and (II), S_w may contain at most $|T_Q(h(X'))|$ simple equations. Therefore, the order of S_w is at most $d + |T_Q(h(X'))|$.

Second, the number of variables, that is the size of the system, is at most $2d + |T_Q(h(X'))|$. The component $2d$ comes from the fact that each border of a constant and/or a variable may introduce in (8) two unknowns, and the component $|T_Q(h(X'))|$ comes from different exponents of (5) for X'. All the other unknowns introduced originally, via simple equations, have disappeared in reduction steps (I) and (II).

Third, the step (I) can increase the absolute values of the coefficients a_{il} in (9). However, the condition $\Sigma_{l=1}^{N_i} |\alpha_{il}| \leq d$ guarantees that their absolute values do not exceed d. Hence, recalling that $c'_i \leq 3d$, the maximal absolute value of the coefficients of the system S_w is at most $3d$. This completes the proof of the lemma. □

PROOF OF THEOREM 19. We first notice that by the proof of Theorem 18 we may assume that $|T_Q(w)| \leq 3d + 4$, where d is the size of the equation e. Next, by Lemma 21, there is a system of linear integer equations S_w such that, for each solution s of S_w, there is a word w' in L such that $T_Q(w')$ is a part of s. Moreover, the parameters of S_w can be chosen as follows

$$o(S_w) \leq 4d + 4, \tag{10}$$

$$s(S_w) \leq 5d + 4, \tag{11}$$

$$d(S_w) \leq 3d. \tag{12}$$

Now let $\mathcal{S}$ be the family of all equations satisfying restrictions (10)–(12). Clearly, $\mathcal{S}$ is finite. Further, let k be the maximal value of the coordinates of all minimal solutions of systems in $\mathcal{S}$. By the König Infinity Lemma, such a finite k exists. Now by Lemma 21, and in particular by (ii), the k thus chosen works for Theorem 18. □

As an application of Theorem 19 we prove a general result that can be used to show that certain relations on words defined only in terms of lengths of words are not expressible. Other applications are given in the next section.

Fix an alphabet Σ. Let f be a function $f: N^r \to N$. We say that *zeros of f are expressible*, or briefly that f is *0-expressible*, if the relation

$$\{(w_1, \ldots, w_r) : f(|w_1|, \ldots, |w_r|) = 0\} \subseteq \Sigma^r$$

is expressible.

THEOREM 22. *Let $|\Sigma| \geq 2$. For any 0-expressible function $f: N^r \to N$, there exists a constant $k = k(f)$ such that if*

$$f(i_1, i_2, \ldots, i_r) = 0,$$

for some $i_1, \ldots, i_r$, and $i_s > k$, for some $s \in \{1, \ldots, r\}$, then also

$$f(i_1, i_2, \ldots, i_{s-1}, p, i_{s+1}, \ldots, i_r) = 0,$$

for some $p < k$.

PROOF. It is enough to prove the result for $s = 1$, that is, for $i_s = i_1$.

Assume first that $i_1 = 3l + 2$ for some $l \geq 1$. Let f be 0-expressible by an equation e via variables $X_1, \ldots, X_r$. Note that e may contain some other variables as well. We introduce two new variables U and Z, and define the following system of equations

$$S: \begin{cases} e, & \\ aabU = Uaab, & \\ X_1 = b^2U, & \\ bX_i = X_ib, & \text{for } 2 \leq i \leq r, \\ Z = X_1abX_2abX_3ab \cdots abX_r. & \end{cases}$$

It is straightforward to see that S expresses via Z the language

$$L = \{b^2(aab)^l(ab)b^{i_2}(ab)b^{i_3} \cdots (ab)b^{i_r} : f(3l + 2, \ldots, i_r) = 0\}. \tag{13}$$

Observe here that, since the set of words $\{aab, ab, b\}$ is a code, cf. Berstel and Perrin [1985], each word u in L admits a unique representation in form

$$b^2(aab)^l(ab)b^{i_2}(ab)b^{i_3} \cdots (ab)b^{i_r}.$$

By Theorem 19, for $Q = aab$, there is a constant k such that if $exp_{aab}(w) > k$, then there is a word u in L with $exp_{aab}(u) \leq k$ which is obtained from w by removing some occurrences of aab. Consider the word

$$w = b^2(aab)^l(ab)b^{i_2}(ab)b^{i_3} \cdots (ab)b^{i_r},$$

with $l - 2 > k$. Clearly, $exp_{aab}(w) = l - 2 > k$. Since the word aab occurs in w only as a subword of $(aab)^l$, there exists in L a word u of the form

$$u = b^2(aab)^p(ab)b^{i_2}(ab)b^{i_3} \cdots (ab)b^{i_r} \tag{14}$$

for some $p \leq k + 2$. Since $u \in L$ and the representation (14) is unique, we have

$$f(3p + 2, i_2, \ldots, i_r) = 0.$$

So in this case we can choose $k(f) = 3(k + 2) + 2$.

The cases $i_1 = 3l + 1$ and $i_1 = 3l$ can be proved in the very same way by replacing $b^2(aab)^l$ in the definition of L by $b^1(aab)^l$ and $b^0(aab)^l$, respectively. □

It is worth noticing that we did not specify precisely how large the constant k in the corollary is. Only if we would formulate the corollary by assuming that f is 0-expressible by an equation of size d, then we could give a bound for $k(f)$.

6. *Applications*

In this section, we show how to apply each of our five results, Theorems 16, 17, 18, 19 and 22, to achieve our original goal: to prove that several very natural properties of words are not expressible. To our knowledge no such result is known in literature except that the property "X being a prefix of Y" cannot be expressed without using additional unknowns, cf. Siebert [1992].

Example 23. The language $L_1 = (a \cup b)^*$ over three-letter alphabet $\Sigma = \{a, b, c\}$ is not expressible. Let $\mathcal{F}$ be the $\mathcal{F}$-factorization defined by factoring words into the blocks of the same letters. Clearly, $\mathcal{F}$ is synchronizing. Assume that the language $L_2 = (a \cup b)^*$ is expressible and let k be a constant from Theorem 16. Consider now the word $w = a^{k+1}ba^kb \ldots ab$ which admits the $\mathcal{F}$-factorization $a^{k+1}, b, a^k, b, \ldots, b, a, b$. Since $n_{\mathcal{F}}(w) = k + 2 > k$, by Theorem 16, there is a pattern $p(X)$ such that $p(\Sigma^*) \subseteq L_1$. Substituting c for X we obtain a contradiction.

Example 24. The language $L_2 = \{w : w \text{ is primitive}\}$ is not expressible. As in the previous example we consider the $\mathcal{F}$-factorization defined by factoring words into the blocks of the same letters, but now we apply Theorem 17. Let k be a constant from Theorem 17. Again consider the word $w = a^{k+1}ba^kb \ldots ab$. Since $n_{\mathcal{F}}(w) = k + 2$, by Theorem 17, there is a pattern $p(X, Y)$ with two variables such that the pattern p synchronizes with the $\mathcal{F}$-factorization of w and $p(\Sigma^*, \Sigma^*) \subseteq L$. Since the pattern p synchronizes with the $\mathcal{F}$-factorization of w, one of the variables, say X, corresponds to a factor of w of the form a^i. Since each factor of this form occurs in w exactly once, the variable occurs exactly once in the pattern. Now we put $Y = \epsilon$ obtaining a pattern with one variable X which occurs in it in only one place, that is, a pattern in the form w_1Xw_2. Setting $X = w_2w_1$, we obtain a contradiction.

Example 25. The language

$$L_3 = \{w : \text{the number of } a\text{'s and } b\text{'s is the same in } w\}$$

over the alphabet $\{a, b\}$ is not expressible. Assume that L_3 is expressible and k is the constant in Theorem 18. Consider the word $a^{k+2}b^2a^{k+1}b^3 \cdots a^2b^{k+2}$. Since $|T_a(w)| = k + 1 > k$, there is u in L_3 which is obtained from w by removing some blocks of a's. This means however that the number of a's in u is

smaller than that in w and the number of b's is the same. Thus, u is not in L_3, a contradiction.

Example 26. The language $L_4 = \{a^n b^n : n \geq 1\}$ is not expressible. We prove it by a contradiction applying Theorem 19, with $Q = a$. Let k be a constant from Theorem 19. Take a word $w = a^{k+1}b^{k+1}$. Since $w \in L_4$ and $exp_a(w) > k$ there exists a word u in L_4, which is obtained from w by removing some occurrences of the word a, a contradiction.

Example 27. The relation "x and y are of equal length", that is,

$$\mathcal{T} = \{(x, y) \in \Sigma^* \times \Sigma^* : |x| - |y| = 0\}$$

is not expressible. This is due to Theorem 22. Observe here, that the relation $\mathcal{T}$ is expressible if $|\Sigma| = 1$. Indeed, it is expressible by the formula

$$X = Y \qquad \text{and} \qquad Xa = aX.$$

As a consequence of the above examples, we easily obtain.

THEOREM 28. *The family of expressible languages is not closed under the operations of complementation, morphic image, inverse morphic image and shuffle.*

PROOF. The nonclosure under complementation follows from Examples 23 and 1. The nonclosure under morphic image (even under length preserving morphisms) is taken care by Example 23, the language $\{a, b, c\}^*$ over $\{a, b, c\}$ and the morphism h defined by $h(a) = h(b) = a$ and $h(c) = b$. Similarly, the language a^* together with the morphism h' defined by $h'(a) = h'(b) = a$ and $h'(c) = c$ shows that expressible languages are not closed under inverse morphisms. Finally, since shuffle of a^* and b^* is $(a \cup b)^*$ we have the nonclosure under the shuffle. □

We conclude this section by emphasizing that the combination of closure and nonclosure properties of the family $\mathcal{L}(\Sigma)$ of expressible languages, especially the closure under intersection and union and the nonclosure under complementation and morphic image, makes the family quite different from usually considered families of formal languages. As a concrete illustration, we give the following result.

THEOREM 29. *The family $\mathcal{L}(\Sigma)$ is a proper subset of the family of recursive languages over Σ, but is incomparable with the families of D0L, regular and context-free languages over Σ.*

PROOF. That languages in $\mathcal{L}(\Sigma)$ are recursive follows from Makanin's algorithm, cf. Makanin [1977], and the strictness of the inclusion, for example, from Example 24.

The language $(a \cup b)^*$ is an example of a regular, and hence of a context-free, language which is not in $\mathcal{L}(\Sigma)$, and not a D0L language either. As in Example 26 one can show that the language $\{a^{2^n}b^{2^n} : n \geq 1\}$ is an example of a D0L language which is not in $\mathcal{L}(\Sigma)$. On the other hand the language of all squares is expressible by the equation $X = ZZ$ and it is neither a context-free nor a D0L language. □

7. *Concluding Remarks*

As a major contribution of this paper, we introduced—according to our knowledge—first tools to show that certain properties of words are not expressible as solutions of word equations, or more precisely as values of some components of solutions of word equations. Our tools were based on special factorizations of words, which we called synchronizing.

We emphasize that there remains a lot of research to be done on this interesting and fundamental field. We point out here just a few open problems:

—*Problem* 1. Is the relation "u is a sparse subword (subsequence) of v" expressible?

—*Problem* 2. Are the properties "w is k-power free", for a fixed $k > 0$, and "w is a Fibonacci word" expressible? Recall, that Fibonacci words are defined by the recurrence formula

$$w_0 = a,\ w_1 = b,\ w_{n+2} = w_{n+1}w_n,\ \text{for } n \geq 2.$$

—*Problem* 3. When is the complement of an expressible language expressible?

—*Problem* 4. Can a gap theorem be formulated for languages expressible by word equations with more than two variables?

—*Problem* 5. Is union of two expressible languages (or relations) expressible by using only one additional variable?

ACKNOWLEDGMENTS. We would like to thank Lucian Ilie for careful reading of our paper and pointing out to us Example 15, and the anonymous referees for their useful comments which improved the presentation.

REFERENCES

ANGLUIN, D. 1979. Finding patterns common to a set of strings. In *Proceedings of 11th Annual Symposium on the Theory of Computing (STOC'79)* (Atlanta, Ga., Apr. 30–May 2). ACM, New York, pp. 130–141.

BERSTEL, J., AND PERRIN, D. 1985. *Theory of Codes*. Academic Press, Orlando, Fla.

BÜCHI, R., AND SENGER, S. 1986/87. Coding in the existential theory of concatenation. *Arch. Math. Logik, 26*, 101–106.

BULITKO, V. K. 1970. Equations and inequalities in a free group and a free semi-group. *Tul. Gos. Ped. Inst. Ucen. Zap. Mat. Kafedr. Geometr. i Algebra 2*, 242–252, (in Russian).

CHOFFRUT, C., AND KARHUMAKI, J. 1997. Combinatorics of words. In *Handbook of Formal Languages*, G. Rozenberg and A. Salomaa, eds. Springer-Verlag, New York, pp. 329–438.

CULIK, K., II, AND KARHUMAKI, J. 1983. Systems of equations and Ehrenfeucht's conjecture. *Discr. Math. 43*, 139–153.

EYONO OBONO, S., GORALCIK, P., AND MAKSIMENKO, M. 1994. Efficient solving of the word equations in one variable. In *Proceedings of MFCS'94*. Lecture Notes in Computer Science, vol. 841. Springer-Verlag, New York, pp. 336–341.

HARRISON, M. A. 1978. *Introduction to Formal Language Theory*. Addison-Wesley Publishing Company, Reading, Mass.

JIANG, T., SALOMAA, A., SALOMAA, K., AND YU, S. 1995. Decision problems for patterns, *J. Comput. Syst. Sci. 50*, 53–63.

KARHUMÄKI, J., MIGNOSI, F., AND PLANDOWSKI, W. 1997. The expressibility of languages and relations by word equations. In *ICALP'97*. Lecture Notes in Computer Science, vol. 1256. Springer-Verlag, New York, pp. 98–109.

KHMELEVSKI, YU. I. 1971/1976. Equations in free semigroups, *Trudy Mat. Inst. Steklov, 107*. (English translation: *Proc. Steklov Inst. of Mathematics 107 (1971)*, American Mathematical Society, Providence, R.I., 1976.)

KOSCIELSKI, A., AND PACHOLSKI, L. 1996. Complexity of Makanin's algorithm, *J. ACM 43*, 4, 670–684.

LENTIN, A. 1972. *Equations dans des Monoides Libres*. Gouthiers-Villars, Paris, France.

LOTHAIRE, M. 1983. *Combinatorics on Words*. Addison-Wesley, Reading, Mass.

MAKANIN, G. S. 1977. The problem of solvability of equations in a free semigroup. *Mat. Sb. 103*, (145), 147–233. (English transl. in *Math. U.S.S.R. Sb. 32*, 1977.)

MATIJASEVICH, Y. 1970/1971. Enumerable sets are diophantine. *Soviet. Math. Doklady 11*, 354–357. (English translation in *Dokl. Akad. Nauk SSSR 191*, 279–282, 1971.)

ROZENBERG, G., AND SALOMAA, A. 1980. *The Mathematical Theory of L Systems*, Academic Press, Orlando, Fla.

SEIBERT, S. 1992. Quantifier hierarchies and word relations. In *Lecture Notes in Computer Science*, vol. 626. Springer-Verlag, New York, pp. 329–338.

RECEIVED DECEMBER 1997; REVISED APRIL 1999; ACCEPTED MAY 1999