

A Constructive Solution of the Language Inequation

$$XA \subseteq BX$$

Olivier Ly

LaBRI – Bordeaux 1 University

Abstract. We consider the inequation $XA \subseteq BX$ where A , B and X are formal languages, X is unknown. It has been proved in [9] that if B is a regular language then the maximal solution is also regular. However, the proof, based on Kruskal's Tree Theorem, does not give any effective construction of the solution. Here we give such an effective construction in the case where A and B are both finite and are such that $\max_{b \in B} |b| < \min_{a \in A} |a|$. Moreover, the complexity of our construction is elementary.

Keywords. Automata, Games, Language Equations

Introduction

Language equations arise in a natural way in computer science. Let us just think about Arden's lemma for instance, or context-free languages which are components of the least solutions of systems of polynomial equations.

However, even very simple questions may appear very difficult. For instance, one can think about the equation $XL = LX$ where X is unknown; this is the long-standing Conway problem which asks whether the maximal language commuting with a given rational language is also rational or not ([2], see also [6, 1, 4, 5]).

Many advances have been done in this domain this last few years ([6, 4, 5]). But Conway's problem has got a solution very recently, actually a negative solution. It has been proved in [11] (see also [3]) that there exists a *finite* language L such that the maximal solution of $XL = LX$ is not recursively enumerable even for some finite A and B . In addition, many natural classes of formal languages have got characterizations in terms of equations (see [12, 13]).

In [9], it has been proved that the maximal solution of $XA \subseteq BX$ is regular if B is regular, whatever A is. But the situation is tight: if one imposes to X to be contained in some given star-free language, then the maximal solution of $XA \subseteq BX$ can become non recursively enumerable (see [8]). This is a variation of the negative result of [11]. Besides, the proof that the maximal solution of $XA \subseteq BX$ is regular is based on the Kruskal's Tree Theorem (see [7]). It is

non constructive, i.e., it does not give any effective construction of the maximal solution.

In this article, we give such an *effective construction* in the case where A and B are both finite and are such that $\max_{b \in B} |b| < \min_{a \in A} |a|$: we set up an algorithm to construct an automaton recognizing the maximal solution of $XA \subseteq BX$. Moreover, the complexity of our algorithm is elementary.

Like in [9], our proof takes the point of view of games. We consider a game with two players: the attacker and the defender. Positions of the game are words. The game consists of a succession of turns as follows: first, the attacker chooses a word $a \in A$ and appends it to w , where w is the current position of the game. If $w.a$ has no prefix in B then the attacker wins and the game stops. Otherwise the defender chooses a prefix of $w.a$ which belongs to B , and cuts it from $w.a$, driving the game to a new position $b \setminus w.a$ for next turn. The defender wins if the game consists of infinitely many turns. Membership of the maximal solution of $XA \subseteq BX$ can be translated into the existence of a winning strategy for the defender (see [9]).

The main ingredient of our proof is a shrinking lemma for words having a winning strategy, it shall be detailed in the text (see Section 2). The hypothesis on lengths of words of A and B is used only for it.

The author wants to thank Professor G. Sénizergues for very helpful discussions; and the anonymous referee for indicating large simplifications of the proof.

Preliminaries

In all the paper, Σ is a finite alphabet. A and B are finite languages over Σ such that

$$\max_{b \in B} |b| < \min_{a \in A} |a|$$

Let w be a word, we denote by $|w|$ the length of w . Let v be a prefix (respectively a suffix) of w . We denote by $v \setminus w$ (respectively w/v) the unique word v' such that $w = vv'$ (respectively $w = v'v$).

The set of finite sequences of elements of A is denoted by T_A , this is the complete A -deterministic tree. “ A -deterministic” because each node has one and only one son associated to each $a \in A$; the edge associated to this son can be considered as a -labeled. The empty sequence, i.e., the root, is denoted by ρ .

1 The Game of $XA \subseteq BX$

In a classical way, the equation $XA \subseteq BX$ can be translated into the game framework as follows.

We are supposed to be given with two languages A and B . We consider a game with two players: *Attacker* and *Defender*. The game consists in a possibly infinite sequence of turns. At the beginning of each turn, the position of the game is a word. One turn on a position w goes as follows:

1. *Attacker* chooses a word $a \in A$ and appends it to the right of w .
2. If no prefix of $w.a$ does belong to B , then *Defender* loses, the game stops and *Attacker* wins. Otherwise, *Defender* chooses a prefix b of $w.a$ belonging to B and erases it from $w.a$. And the game continues in the position $b \setminus w.a$.

So, *Attacker* wins if he manages to block *Defender*; and *Defender* wins if the game consists of an infinite number of turns in which he never loses. We say that *Defender* has a winning strategy on a word w if he has a strategy which makes the game starting on w continue forever whatever *Attacker* does.

Lemma 1 (Equation and Game). *A word w belongs to the maximal solution of $XA \subseteq BX$ if and only if *Defender* has a winning strategy.*

Proof. See [10].

2 A Shrinking Lemma on *Attacker*'s Strategies

2.1 The Shrinking Lemma

Let us denote by S the set of strict prefixes of words of B , including the empty word. Let us consider a word w . We say that some $s \in S$ is *accessible through w* by *Defender* if w can be written as $b_1 \dots b_n s$ where all the b_i 's are words of B . The set of all elements of S which are accessible through w is called the *visibility of *Defender* through w* . It is denoted by $\text{Vis}(w)$.

Remark 1. If $\text{Vis}(w) = \emptyset$ then *Attacker* has a winning strategy on w . But the converse is false. Such a w is said to be *terminal*.

Definition 1 (B-relation). *Let us be given with two words w and w' . We say that w and w' are B -related, which is denoted by $w \leftrightarrow_B w'$, if there exist 4 words v_1, v_2, v_3 and v'_2 such that:*

- $w = v_1 v_2 v_3$ and $w' = v_1 v'_2 v_3$.
- For any $s \in \text{Vis}(v_1)$, $\text{Vis}(s v_2) = \text{Vis}(s v'_2)$.
- $|v_1| \geq N_1$.

where we define

$$N_1 = \lceil 2 \frac{(\min_{a \in A} |a|)(\max_{b \in B} |b|)}{\min_{a \in A} |a| - \max_{b \in B} |b|} (2^{|S|^2} + 1) \rceil$$

Let us note that v_3 is superfluous in this definition. We just keep it for convenience of notations.

Remark 2. For any two words w and w' , $w \leftrightarrow_B w'$ implies that $\text{Vis}(w) = \text{Vis}(w')$.

Lemma 2. *The relation \leftrightarrow_B is a right congruence of finite index over the set of words of length greater than N_1 .*

Proof. To see that \leftrightarrow_B is a right congruence, only transitivity is not straightforward: Let $w \leftrightarrow_B w'$ and $w \leftrightarrow_B w''$. Let us note according to the previous notations:

- $w = v_1v_2$ and $w' = v_1v'_2$ (here we omit v_3 which is supposed to be added at the right of v_2).
- $w' = \bar{v}_1\bar{v}_2$ and $w'' = \bar{v}_1\bar{v}'_2$

Let us assume that $|\bar{v}_1| > |v_1|$. Therefore, v_1 is a prefix of w'' . Let us pick some $s \in \text{Vis}(v_1)$. Then $\text{Vis}(sv_2) = \text{Vis}(sv'_2)$. Let $\bar{S} = \text{Vis}(s.(v_1 \setminus \bar{v}_1))$. Then $\text{Vis}(sv'_2)$ is the union of all the $\text{Vis}(\bar{s}\bar{v}_2)$ for $\bar{s} \in \bar{S}$. Besides, any $\bar{s} \in \bar{S}$ belongs also to $\text{Vis}(\bar{v}_1)$. And so, by assumption, $\text{Vis}(s)\bar{v}_2 = \text{Vis}(s)\bar{v}'_2$. Thus, $\text{Vis}(sv_2)$ is the union of all the $\text{Vis}(\bar{s}\bar{v}'_2)$ for $\bar{s} \in \bar{S}$, which is in turn equal to $\text{Vis}(s(v_1 \setminus \bar{v}_1).\bar{v}'_2)$. This proves that w and w'' are B-related with $v'_2 = (v_1 \setminus \bar{v}_1).\bar{v}'_2$.

To verify that \leftrightarrow_B is of finite index, it suffices to note that any sufficiently large word is equivalent to a shorter word, prefix of it, and of bounded length. To get that, one uses a simple counting argument, based on the fact that S is a finite set and therefore has a finite number of subsets.

Let us mention that the congruence is computable.

Let σ be a strategy for *Attacker* (respectively *Defender*) over a word w . A *finite σ -sequence of plays* in the game is a finite sequence of plays $(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)$ where *Attacker* (respectively *Defender*) has played according to σ . This means that each a_i of the sequence has been chosen according to the previous b_j for $j < i$ and σ . The b_i are unspecified and variable.

A *strong strategy* for *Attacker* (respectively *Defender*) is a strategy in the game modified in such order that *Attacker* (respectively *Defender*) can play several words of A (respectively B) at the same turn. Formally, a strong strategy for *Attacker* (respectively *Defender*) is a strategy in the game defined by the pair (A^+, B) (respectively (A, B^+)) instead of (A, B) . Let us note that if *Attacker* has a winning strong strategy, then he has a winning strategy, and this is the same thing for *Defender*. Indeed, provided with a winning strong strategy, one can win in the normal game by maintaining a (FIFO) queue of plays. We just use the concept of strong strategy in order to be more comfortable when describing winning strategies.

Lemma 3 (Shrinking Lemma). *Let us be given with two B -related words w and w' , and a strategy σ for Attacker over w . Then there exists an integer L and a strong strategy σ' for Attacker over w' with the following property: Whatever the plays of Defender, by following σ' , in less than L turns:*

- Either Attacker wins
- Or he drives the game from w' to a new word v' such that there exists a non-void finite σ -sequence of plays driving the game from w to a new word v which is B -related to v' .

The integer L and σ' depend on σ , w and w' .

Before going into its proof, let us state the main consequence of this result:

Theorem 1. *Let w and w' be two B -related words. Then Attacker has a winning strategy over w if and only if he has one on w' .*

Proof. Indeed, let us suppose that Attacker has a winning strategy over w . We can construct a winning strategy over w' as follows:

Lemma 3 provides us a strategy σ' and an integer L . Let Attacker start playing according to this strategy. According to the Lemma, after a finite number of turns, less than L :

- Either Attacker wins. That is what we wanted and σ' stops here.
- Or else, he drives the game to a word v'_1 and the lemma provides us σ -sequence of plays driving the game from w to a new word v_1 which is B -related to v'_1 .

The strategy σ is still winning over this new word v_1 , we then start again the process with the words v_1 and v'_1 . And so on.

Following that, we construct a sequence $v_1, v_2, \dots, v_k, \dots$ of words which are positions of a play in the game where Attacker follows σ . Let us note that each pair v_i, v_{i+1} are separated by at least one turn, and in fact, several turns. In particular, when the game arrives to the word v_k , at least k turns have been played. Besides, let us observe that σ , as a winning strategy of Attacker, is finite. This implies that there exists an integer L_σ such that Attacker wins for sure in less than L_σ turns from w . Therefore, k is also bounded by L_σ . This implies that our process stops after at most L_σ cycles, which means that Attacker wins within at most L_σ cycles.

In the proof of Lemma 3 we need the following concept:

Definition 2 (Waiting Loop). *Let w be a word. We define a waiting loop to be a decomposition $w = w_1w_2w_3w_4$ of w in 4 factors such that for any $s \in \text{Vis}(w_1)$, $\text{Vis}(sw_2) = \text{Vis}(sw_2w_3)$ and w_3 not empty.*

Lemma 4 (Waiting Loops and B-Relation). *Let w be a word, and let $w = w_1w_2w_3w_4$ be a waiting loop such that $|w_1| \geq N_1$. Then w is B-related to any word of $w_1w_2w_3^*w_4$.*

Proof. So, let $w' = w_1w_2w_3^kw_4$ for some integer k . According to the notations of Definition 1, let us define $v_1 = w_1$, $v_2 = w_2w_3$, $v'_2 = w_2w_3^k$ and $v_3 = w_4$. First, let us note that $|v_1| > N_1$ is true, it is an hypothesis of the lemma.

We prove that for any $s \in \text{Vis}(v_1)$, $\text{Vis}(sv_2) = \text{Vis}(sv'_2)$ by induction on k . For $k = 0$, there is nothing to prove: this is the hypothesis of the lemma. Let us thus suppose that it is true for some $k \geq 0$. By induction, we just have to prove that for any $s \in \text{Vis}(v_1)$, $\text{Vis}(sw_2w_3^k) = \text{Vis}(sw_2w_3^{k+1})$.

So let $s \in \text{Vis}(v_1)$. Let $s' \in \text{Vis}(sw_2w_3^k)$. Let us show that $s' \in \text{Vis}(sw_2w_3^{k+1})$. Let $b_1, \dots, b_n \in B$ be such that $b_1 \dots b_n s' = sw_2w_3^k$. Let n' be the greatest index such that $b_1 \dots b_{n'}$ is a prefix of sw_2 . And let $s'' = b_1 \dots b_{n'} \setminus sw_2$; s'' belongs to $\text{Vis}(sw_2)$. Besides, by assumption, $\text{Vis}(sw_2) = \text{Vis}(sw_2w_3)$. Therefore, there exists $b'_1, \dots, b'_{n''}$ such that $b'_1 \dots b'_{n''} \setminus sw_2w_3 = s''$. Finally, we obtain that

$$b'_1 \dots b'_{n''} b_{n'+1} \dots b_n s' = sw_2w_3w_3^k = sw_2w_3^{k+1}$$

which means that $s' \in \text{Vis}(sw_2w_3^{k+1})$. That is what we wanted. The converse is similar.

Existence of waiting loop is given by the following simple result based on a simple counting argument based on the fact that visibility sets are subsets of S .

Lemma 5 (Existence of waiting loops, Version 1). *For any word w and any prefix w_1 of w such that the length of $w_1 \setminus w$ is greater than $2^{|S|^2}$, there exists a waiting loop of form $w = w_1w_2w_3w_4$.*

We actually will use a more precise version:

Lemma 6 (Existence of waiting loops, Version 2). *Let w be a word, and let $w = c_1c_2 \dots c_n$ be a decomposition of w into n factors, where the c_i 's are words. Let n_1 be such that $n - n_1 \geq 2^{|S|^2}$. Then w has a waiting loop $w = w_1w_2w_3w_4$ such that $w_1 = c_1c_2 \dots c_{n_1}$ and the other w_i 's are concatenations of some c_i 's. Formally: for $i = 1, \dots, 4$, $w_i = c_{n_{i-1}+1} \dots c_{n_i}$, where n_0, n_2, n_3 and n_4 are such that $0 < n_1 < n_2 < n_3 \leq n$, $n_0 = 0$ and $n_4 = n$.*

Proof (of Lemma 3).

We describe the strategy σ' from w' turns after turns. In order to do that, we describe a game \mathcal{G}' where *Defender*'s plays are generic, and doing that we describe turns after turns how *Attacker* has to play. During this description, we shall use σ as an *oracle* to which we provide plays of *Defender* and which tells us what σ suggests for *Attacker*'s plays.

First of all, let us observe that *Defender* must play at least $\lfloor \frac{N_1}{\max_{b \in B} |b|} \rfloor$ turns before completely erasing v_1 (here we keep notations of Definition 1 where $w = v_1 v_2 v_3$ and $w' = v_1 v_2' v_3$). And besides, from the definition of N_1 and the fact that $\min_{a \in A} |a| > \max_{b \in B} |b|$ we get:

$$\frac{N_1}{\max_{b \in B} |b|} \geq \frac{N_1}{\min_{a \in A} |a|} + 2(2^{|S|^2} + 1) \quad (1)$$

Let us define

$$N'_1 = \lceil \max_{b \in B} |b| (2^{|S|^2} + 1) \rceil$$

There are 3 stages in the strategy:

1. Informally, the first stage starts at the beginning and goes on until the word obtained by concatenating all plays of *Attacker* is sufficiently long, actually more than $N_1 + N'_1$.

According to the preliminary remark, during this stage, plays of *Defender* remains into v_1 , because v_1 is supposed to be great enough (see below). For these plays thus, there is no difference between w and w' which both have v_1 as a common prefix. Then \mathcal{G}' can be considered as a game \mathcal{G} on w and *Attacker* follows σ .

So, precisely, the first stage consists of the n_1 first plays of the game $(a_1, b_1), (a_2, b_2), \dots, (a_{n_1}, b_{n_1})$ where n_1 is such that $|a_1 a_2 \dots a_{n_1-1}| \leq N_1 + N'_1 < |a_1 a_2 \dots a_{n_1}|$. Let us note that n_1 , i.e., the moment at which Stage 1 ends, depends on the plays of *Defender* and on σ which tells *Attacker* how to play. However, we can say that $n_1 \leq (N_1 + N'_1) / \min_{a \in A} |a| + 1$. This implies that

$$n_1 \leq \frac{N_1}{\min_{a \in A} |a|} + \frac{\max_{b \in B} |b|}{\min_{a \in A} |a|} (2^{|S|^2} + 1) \leq \frac{N_1}{\min_{a \in A} |a|} + 2^{|S|^2} + 1$$

Together with Equation 1 of the preliminary remark, one can conclude that v_1 has not been totally erased, and even more than that: It remains at least $2^{|S|^2} + 1$ turns before that, this means that there exists a word v of length greater than $\max_{b \in B} |b| (2^{|S|^2} + 1)$ such that $v_1 = b_1 b_2 \dots b_{n_1} v$. In particular, this justifies the fact that *Attacker* can use σ to play during this stage.

2. In Stage 2, *Attacker* looks for a *waiting loop*. Formally: *Attacker* plays according to σ until Turn n_2 such that there exists n'_2 such that

$$[w a_1 \dots a_{n_1}] [a_{n_1+1} \dots a_{n'_2}] [a_{n'_2+1} \dots a_{n_2}] \varepsilon$$

is a waiting loop. We choose n_2 to be minimal for this property. Thanks to Lemma 6, because the length of v is greater than $\max_{b \in B} |b| \cdot (2^{|S|^2} + 1)$, we

are sure that n_2 occurs before v has been totally erased. In particular, *Attacker* can still use σ to play during this stage.

3. During Stage 3, *Attacker* no longer follows σ . He plays the sequence $a_{n'_2+1}, \dots, a_{n_2}$ in loop until *Defender* has almost erased v'_2 , i.e., until Turn n_3 which is such that $(b_1 \dots b_{n_3}) \setminus v_1.v'_2 \in \text{Vis}(v_1.v'_2)$ where $b_{n_2+1}, \dots, b_{n_3}$ are all the plays of *Defender* during Stage 3. In the following, $(b_1 \dots b_{n_3}) \setminus v_1.v'_2$ is denoted by s' . Let us note that at this point, *Attacker* may be inside the loop, i.e., he maybe playing some a_i with $n'_2 + 1 \leq i < n_2$. Then whatever the plays of *Defender*, *Attacker* finishes the current loop. This drives the game to some Turn n_4 such that $a_{n_4} = a_{n_2}$.

Let us note that while *Attacker* is finishing his loop, which takes at most $2^{|S|^2} + 1$ turns, it may happen that *Defender* erases v_3 and starts erasing the first plays of *Attacker*, i.e., the plays of Stage 1. However, Stage 1 above ensures that the first n_1 plays of *Attacker* make a word of length greater than $N_1 + N'_1$. Therefore, *Defender* must leave at least a word of length N_1 from $a_1 a_2 \dots a_{n_1}$.

Now, the succession of plays which have been done is

$$(a_1, b_1), (a_2, b_2), \dots, (a_{n_1}, b_{n_1}), \dots, (a_{n'_2}, b_{n'_2}), \dots \\ \dots (a_{n_2}, b_{n_2}), \dots, (a_{n_3}, b_{n_3}), \dots, (a_{n_4}, b_{n_4}).$$

Let n'_3 be the greatest integer such that $n_2 \leq n'_3 \leq n_3$ and $(b_1 \dots b_{n'_3}) \setminus v_1 \in \text{Vis}(v_1)$. In the following $(b_1 \dots b_{n'_3}) \setminus v_1$ is denoted by s . We have that $(b_{n'_3+1} \dots b_{n_3}) \setminus s.v'_2 = s'$. Therefore $s' \in \text{Vis}(s.v'_2)$. Besides, by the definition of \bar{B} -relation, $\text{Vis}(s.v'_2) = \text{Vis}(s.v_2)$. Therefore $s' \in \text{Vis}(s.v_2)$, and thus there exist $\bar{b}_1, \dots, \bar{b}_m$ such that $\bar{b}_1 \dots \bar{b}_m s' = s.v_2$.

Now, let us consider the game \mathcal{G} on w defined as follows: in this game, we consider the sequence of *Defender*'s plays defined by

$$b_1, \dots, b_{n_2}, \dots, b_{n'_3}, \bar{b}_1, \dots, \bar{b}_m, b_{n_3+1}, \dots, b_{n_4}$$

Let us consider the sequence of *Attacker*'s plays corresponding to it according to σ : $\bar{a}_1, \dots, \bar{a}_{m'}$ where $m' = n'_3 + m + n_4 - n_3 + 1$. The first n_2 plays \bar{a}_i 's are exactly the a_i 's that we have been just defined for σ' in Stages 1 and 2, since in these stages, *Attacker* actually used σ .

Let us go back to the definition of σ' on the game \mathcal{G}' . Let us recall that we are at Turn $n_4 + 1$, and *Attacker* is going to play. We define his play to be the concatenation of $\bar{a}_{n_2+1} \dots \bar{a}_{m'}$, let us denote it by a_{n_4+1} . Let us recall that we define σ' as a strong strategy. This means that *Attacker* plays in \mathcal{G}' just like if it were in \mathcal{G} . Let b_{n_4+1} be the next play of *Defender*.

Now, let us observe that if w' has not been totally erased, it remains the same word in \mathcal{G} and in \mathcal{G}' to complete the pass, which is $(b_1 \dots b_{n_4} b_{n_4+1}) \setminus w'$, let us denote it by u .

The description of σ' ends here. To conclude, let us remark that if at some point, *Defender* cannot play any more because the current word has no prefix in B , then *Attacker* wins the game and σ' ends.

It remains to show that the positions of the games \mathcal{G} and \mathcal{G}' are B -related words. To see that, we apply Lemma 4.

Let us suppose that w' has not been totally erased and that it remains the word u defined as above. Altogether, we get for \mathcal{G}' a position of form:

$$\underbrace{ua_1a_2 \dots a_{n_1}}_{w_1} \underbrace{a_{n_1+1} \dots a_{n'_2}}_{w_2} \dots \underbrace{(a_{n'_2+1} \dots a_{n_2}) \cdot (a_{n'_2+1} \dots a_{n_2}) \dots (a_{n'_2+1} \dots a_{n_2})}_{w_3^*} \underbrace{\bar{a}_{n_2+1} \dots \bar{a}'_m}_{w_4} \quad (2)$$

where $\bar{v}'_2 \in a_{n_1+1} \dots a_{n'_2} (a_{n'_2+1} \dots a_{n_2})^+$. In \mathcal{G} we get a position of form:

$$\underbrace{ua_1a_2 \dots a_{n_1}}_{w_1} \underbrace{a_{n_1+1} \dots a_{n'_2}}_{w_2} \underbrace{a_{n'_2+1} \dots a_{n_2}}_{w_3} \underbrace{\bar{a}_{n_2+1} \dots \bar{a}'_m}_{w_4} \quad (3)$$

The words w_1, w_2, w_3 and w_4 defined above satisfy by construction the condition of Lemma 4. They have been indeed chosen just in order to construct a waiting loop. In addition, as we have seen during the description of σ' , if w' has been totally erased, then this part remaining from $a_1a_2 \dots a_{n_1}$ is still of length greater than N_1 .

To conclude the proof, we have to give a bound on the number of turns which have been done. Observe that during the 3 stages of plays, w' can be erased and at most $2^{|S|^2} + 1$ turns can be played after (in order to end the loop). This gives the following bound:

$$L = \frac{|w'|}{\min_{b \in B} |b|} + 2^{|S|^2} + 1$$

3 Effective construction of the solution

To conclude, we consider the right congruence of Lemma 2 over words of length greater than N_1 . It can be extended easily to a right congruence over all words by setting any word of length less than N_1 equivalent to itself only. One gets a new right congruence which still is of finite index. By Theorem 1, any two

congruent words both belong in the greatest solution of $AX \subset XB$ or both do not.

Let us consider an automaton associated to it; and let us select the largest subset of states of this automaton such that if we consider this set as the set of final states; then we get a language which is solution of $AX \subset XB$.

References

1. C. Choffrut, J. Karhumäki, and I. Petre. The Commutation of Finite Sets: A Challenging Problem. *Theoretical Computer Science*, 273:69–79, 2002.
2. J.H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971.
3. J. Karhumäki, M. Kunc, and A. Okhotin. Computing by Commuting. *Theoretical Computer Science*, 356(1-2):200–211, 2006.
4. J. Karhumäki, M. Latteux, and I. Petre. Commutation with Codes. *Theoretical Computer Science*, 340(1), 2005.
5. J. Karhumäki, M. Latteux, and I. Petre. Commutation with Ternary Sets of Words. *Th. Comput. Syst.*, 38(2), 2005.
6. J. Karhumäki and I. Petre. Conway’s Problem for Three-Word Sets. *Theoretical Computer Science*, 289:705–725, 2002.
7. J.B. Kruskal. Well-Quasi-Ordering, the Tree Theorem and Vazsonyi’s Conjecture. *Trans. Amer. Math. Soc.*, 95:210–225, 1960.
8. M. Kunc. On Language Inequalities $XX \subseteq LX$. In *Developments in Language Theory (DLT)*. Springer, 2005.
9. M. Kunc. Regular Solutions of Language Inequalities and Well Quasi-Orders. *Theoretical Computer Science*, 348(2-3):277–293, 2005.
10. M. Kunc. Simple Language Equations. *Bulletin of the European Association for Theoretical Computer Science*, 85:81–102, 2005.
11. M. Kunc. The Power of Commuting with Finite Sets of Words. In *LNCS*, volume 2404, 2005.
12. A. Okhotin. Decision Problems for Language Equations. In *Internat. Colloq. on Automata, Languages and Programming (ICALP)*, pages 239–251. LNCS 2719, 2003.
13. A. Okhotin. A Characterization of the Arithmetical Hierarchy by Language Equations. In *Descriptive Complexity of Formal Systems (DCFS)*, pages 225–237, 2004.