Post Correspondence Problem with Partially Commutative Alphabets

Barbara Klunder[2] * and **Wojciech Rytter**[1,2]**

[1] Institute of Informatics, University of Warsaw, ul. Banacha 2, 02-097 Warszawa, Poland
[2] Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Toruń, Poland

Abstract. We introduce a version of Post Correspondence Problem (PCP, in short) generalized to words over partially commutative alphabets. Several observations are presented about the algorithmic status of the introduced problem. In particular solvability is shown for the partially commutative PCP for two special cases: the binary case of PCP (denoted by PCP(2)), and the case with one periodic morphism. This extends solvability results for the classical PCP for these cases. Also a weaker version of PCP, named here Weak-PCP, is discussed. This version distinguishes (in the sense of solvability) the case of noncommutative from the case of partially commutative alphabets. We consider also a solvable (though NP-hard) simple version of Weak-PCP. Our solvability results demonstrate the power of Ibarra's algorithms for reversal bounded multi-counter machines.

Keywords: Post Correspondence Problem, morphism, partially commutative alphabet, solvability, equality set, weak equality set, reversal bounded multicounter machine

1 Introduction

The origins of partial commutativity is the theory of traces (i.e. monoids with partial commutations). In the fundamental Mazurkiewicz's paper [20], trace languages are regarded as a powerful means for description of behaviors of concurrent systems. The formal language theory over traces, limited to recognizable and rational trace languages, is the subject of [21].

Usually traces are more complicated than standard noncommutative words, for example rational expressions with classical meaning are less powerful then expressions for alphabets which are partially commutative. In the theory of traces

* Supported by grant N N206 258035 of the Polish Ministry of Science and Higher Education.
** Supported by grant N206 004 32/0806 of the Polish Ministry of Science and Higher Education.

the symbols represent some atomic processes and two symbols commute iff they are concurrent (the corresponding processes can be executed in any order).

A partially commutative alphabet (p.c. alphabet, in short) is a finite set A of symbols with a relation $\mathcal{I} \subseteq A \times A$ which is symmetric and irreflexive.
Such a relation is named the *independency relation* or the relation of *partial commutativity*. The complement $\mathcal{D}$ of $\mathcal{I}$ is named the *dependency relation*.

For a given p.c. alphabet A and two words $x,\ y$ we write $x\ \approx_I\ y$ iff the word x can be transformed to y commuting neighboring symbols which are in the relation $\mathcal{I}$. In other words x, y are equivalent modulo pairs of adjacent symbols which commute.

Example. Let $A\ =\ \{a, b, c\}$ and $\mathcal{I}\ =\ \{(a, b),\ (b, a)\}$, then

$$aaabbcab\ \approx_I\ bbaaacba.$$

1.1 The classical Post Correspondence Problem

The Post Correspondence Problem (PCP, in short) is probably the first published algorithmically undecidable combinatorial problem. It is a useful tool to establish undecidability results in different fields of theoretical computer science.

As an example let us recall the problem of emptiness of intersection of two context-free languages.
The PCP problem with lists of size n (denoted by PCP(n)), is defined as follow.:

Given two lists of words over an alphabet A:

$$(u_1, u_2, \ldots, u_n),\ (v_1, v_2, \ldots, v_n)$$

decide if there exists a nonempty sequence $i_1, \ldots i_m$ of indices such that

$$u_{i_1} \ldots u_{i_m} = v_{i_1} \ldots v_{i_m}.$$

Equivalently, for an n-element alphabet X we are given two morphisms

$$h, g : X^\star \mapsto A^\star\ ,$$

and the problem is to decide whether the following set, called the *equality set*, is nonempty:

$$\text{EQ-SET}(h, g) = \{w \in X^+ : h(w) = g(w)\}.$$

1.2 Post Correspondence Problem with p.c. alphabets

In the case of p.c. alphabet we define the equality set with respect to the relation $\mathcal{I}$ of partial commutation:

$$\text{EQ-SET}_I(h, g) = \{w \in X^+ : h(w)\ \approx_I\ g(w)\}$$

Now the partially commutative PCP problem is defined as follows:

given h, g and an independency relation $\mathcal{I}$,
check if EQ-SET$_I(h, g) = \emptyset$.

The only known (positive) result related to PCP with commutative alphabet is of [13] and deals with the Parikh equivalence, i.e. the case when the p.c. alphabet is fully commutative.
It is known that the classical PCP is solvable for the lists of size $n = 2$ and unsolvable for $n \geq 7$, the case of $2 < n < 7$ is not well understood.
For partially commutative PCP the situation is similar.

1.3 Reversal bounded multicounter machines

As an algorithmic tool (to show solvability) we use the algorithm testing emptiness of reversal bounded multicounter machines. In this subsection we define these machines and state the basic result needed later. A two-way k-counter machine M is defined by an 8-tuple $M =< k, K, \Sigma, \$, \bar{c}, \delta, q_0, F >$ where:

1. $K, \Sigma, \bar{c}, \$, q_0, F$ are the states, inputs, left and right endmarkers, initial and final states respectively;
2. δ is a mapping from $K \times (\Sigma \cup \{\bar{c}, \$\}) \times \{0, 1\}^k$ into $K \times \{-1, 0, 1\} \times \{-1, 0, 1\}^k$.

Assume the counters cannot be decreased below zero.
A configuration $\bar{c}x\$$, $x \in \Sigma$, is defined by a tuple $(q, \bar{c}x\$, i, c_1, \ldots, c_k)$, denoting that M is in the state q with the head reading the i-th symbol of $\bar{c}x\$$ and $c_1, \ldots, c_k$ are integers stored in the k counters.
We define a relation $\rightarrow$ among configurations as follows:

$$(q, \bar{c}x\$, i, c_1, \ldots, c_k) \rightarrow (p, \bar{c}x\$, i + d, c_1 + d_1, \ldots, c_k + d_k)$$

if a is the ith symbol of $\bar{c}x\$$ and $\delta(q, a, \lambda(c_1), \ldots, \lambda(c_k))$ contains $(p, d, d_1 \ldots, d_k)$, where

$$\lambda(c_i) = \begin{cases} 0 & \text{if } c_i = 0 \\ 1 & \text{if } c_i \neq 0 \end{cases}$$

The reflexive and transitive closure of $\rightarrow$ is denoted as $\rightarrow^*$. A string x in Σ^* is accepted by M if

$$(q_0, \bar{c}x\$, 1, 0, \ldots, 0) \rightarrow^* (q, \bar{c}x\$, i, c_1, \ldots, c_k),$$

for some $q \in F$, $1 \leq i \leq |\bar{c}x\$|$ and nonnegative integers $c_1, \ldots, c_k$. The set of strings accepted by M is denoted by $L(M)$. A *reversal-bounded* k-counter machine operates in such a way that in every accepting computation the input head reverses direction at most p times and the count in each counter alternately increases and decreases at most q times, where p, q are some constants.
The emptiness problem for M is to check if $L(M) = \emptyset$. We will use one of the results of Ibarra's paper.

Lemma 1. [12]
The emptiness problem for reversal-bounded multicounter machines is solvable.

2 Two special cases of partially commutative PCP

For a pair of symbols (a, b) we denote by $\pi_{a,b}$ the projection which for a word w removes all letters but a, b.

Example. $\pi_{a,b}(accbacb) = abab$, $\pi_{a,c}(accbacb) = accac$.

The following result reduces the relation $\approx$ to multiple application of equality of classical strings over noncommutative alphabet.

Lemma 2. [8] $u \approx_I w \Leftrightarrow (\forall (a,b) \notin \mathcal{I}) \ \pi_{a,b}(u) = \pi_{a,b}(w)$.

According to Lemma 2 we can express the equality set for PCP with p.c. alphabets as a finite intersection of equality sets for standard (noncommutative) alphabets.

Lemma 3. $EQ\text{-}SET_I(h, g) = \bigcap_{(a,b)\notin\mathcal{I}} EQ\text{-}SET(\pi_{a,b} \cdot h, \pi_{a,b} \cdot g)$.

2.1 Partially commutative PCP(2).

In this section we assume that $n = 2$ and $X = \{0, 1\}$. The problem PCP(2) is solvable as was proved by Ehrenfeucht, Karhumäki and Rozenberg in 1982, see [3]. On the other hand Matiyasevich and Sénizergues showed that PCP(7) is unsolvable, [19]. Before we state the first new result we shall recall the main results of [4] and [9], which can be found in [8] too, concerning the structure of equality sets in the case of free monoids.

We say that a morphism $h : X^\star \mapsto A^\star$ is periodic if $h(X) \subseteq u^\star$ for some word u. For a symbol s by $|w|_s$ denote the number of occurrences of s in w.

Lemma 4. ([4])
(a) *If h and g are periodic then either*
$EQ\text{-}SET(h, g) = \emptyset$ *or* $EQ\text{-}SET(h, g) = \{w \in X^\star : r(w) = \frac{|w|_0}{|w|_1} = k\}$
for some $k \geq 0$ or $k = \infty$.
(b) *If h is periodic and g is not then $EQ\text{-}SET(h, g)$ is empty or equal to u^+ for some nonempty word u.*

Hence equality sets are regular or accepted by a reversal bounded one-counter machines. The number $r(w) = \frac{|w|_0}{|w|_1}$ is called the *ratio* of a word and it is decidable if the intersection of regular sets and (or) sets of words of a given ratio is nonempty.

In the case of two non-periodic morphisms the equality set is always regular. For two periodic morphisms, EQ-SET(h, g) can be nonempty only when $h(X), g(X) \subseteq u^\star$ and then r can be easily found. The main result of [10] says that in the nonperiodic case the equality set is of a very simple form.

Lemma 5. [10]
Let (h, g) be a pair of non-periodic morphisms over a binary alphabet. If the equality set EQ-SET(h, g) is nonempty then it is of the form $(u + v)^+$ for some words u, v.

The following constructive result has been shown in [9] (as Corollary 5.7).

Lemma 6. [9]
Assume the size of the lists is $n = 2$ and h, g are two nonperiodic morphisms. Then EQ-SET(h, g) can be effectively found (as a regular expression or finite automaton).

A combination of these results implies easily the following fact.

Theorem 1.
Partially commutative PCP(2) is solvable

Proof.
Lemmas 2,3,4,5 imply that the p.c. PCP(2) can be reduced to the emptiness of intersection of a finite set of languages, each of them is regular or a reversal bounded one-counter language. These languages can be effectively presented by corresponding finite automata or reversal bounded one-counter automata. Hence the intersection language can be accepted by a reversal-bounded multicounter machine M which is a composition of all these automata. Due to Lemma 1 the emptiness problem for M is solvable. Csonsequently the emptiness of the intersection of related languages is also solvable. This completes the proof.

2.2 Partially commutative PCP with one periodic morphism

We shall consider here another easily solvable case: periodic morphisms. We say that a morphism h into a p.c. alphabet is periodic if there is a word w such that for each x $h(x) \approx_I w^i$ for some natural i. The proof of the next theorem adopts similar arguments as the classical one, see [8].

Theorem 2.
Partially commutative PCP is decidable for instances (h, g), where h is periodic.

Proof.
Let $h, g : X^\star \mapsto A^\star$ and assume that h is periodic Let $(a, b) \in \mathcal{D}$, then the equality set of $(\pi_{a,b} \cdot h, \pi_{a,b} \cdot g)$ is a multicounter reversal bounded language.

Now the equality set of (h, g) is the intersection of multicounter reversal bounded languages too. Define the morphism ρ by $\rho(a) = |h(a)| - |g(a)|$ for all $a \in X$. Define also the set $R = g^{-1}(u^\star \setminus \{\varepsilon\})$. We have:

1. $\rho^{-1}(0) = \{v : |h(v)| = |g(v)|\}$
2. $w \in \rho^{-1}(0) \cap R \Leftrightarrow w \neq \varepsilon, g(w) \in u^\star$ and $|g(w)| = |h(w)|$.

Hence $g(w) = h(w)$ for some $w \neq \varepsilon$ if and only if $\rho^{-1}(0) \cap R \neq \emptyset$. The language $\rho^{-1}(0) \cap R$ is recognizable by a reversal-bounded multicounter machine. Hence emptiness is solvable due to [12]. This completes the proof.

3 Partially commutative weak PCP

There is a version of PCP which is easily solvable for noncommutative alphabets but surprisingly the same version is unsolvable for partially commutative alphabets.

Define the partially commutative problem, named here *Weak* PCP, with parameters r, s as follows:

given p.c. words $x_1, x_2, \ldots x_r,\ y_1, y_2, \ldots y_s$,
test if there are nonempty sequences

$$(i_1, i_2, \ldots, i_p),\ (j_1, j_2, \ldots j_q)$$

such that

$$x_{i_1} x_{i_2} \ldots x_{i_p} \quad \approx_I \quad y_{j_1} y_{j_2} y_{j_q}.$$

We can redefine it using the concept of equality sets as follows:

given two morphisms h, g into p.c. words, test emptiness of the set

$$\text{Weak-EQ-SET}(h, g) = \{(z1, z2)\ :\ h(z1)\ \approx_I\ g(z2)\}.$$

The set Weak-EQ-SET(h, g) is called here the weak equality set. If we do not write *partially commutative* this means that we consider the case of classical noncommutative alphabet (special case of partially commutative).

Observation. The language $\{x\#y^R\ :\ (x, y) \notin \text{Weak-EQ-SET}(h, g)\ \}$ is a linear context free language.

Proof.
The one-turn pushdown automaton can nondeterministically guess two symbols $a \neq b,\ (a, b) \notin I$. Then it reads x and writes on the stack the word $\pi_{a,b}(x)$, then it checks, while reading y^R that $\pi_{a,b}(x) \neq \pi_{a,b}(y)$.

Denote by Weak-PCP(s, r) the weak PCP in which the domain of h is of size s and the domain of g is of size r.

The natural relation of PCP for noncommutative alphabets to Weak-PCP for p.c. alphabets is as follows.
Assume we have an instance of PCP given by (u_i, v_i) for $i = 1 \ldots k$, where $u_i, v_i \in A^+$ and $A \cap \{1, \ldots, k\} = \emptyset$.

Let the p.c. alphabet be $A \cup \{1, \ldots, k\}$, where all letters in A commute with all letters in $\{1, \ldots, k\}$, and no other pairs of different letters commute.

Define

$$h(i) = i \cdot u_i,\ g(i) = i \cdot v_i \ \text{ for each } 1 \leq i \leq k$$

Then we can express in a natural way the PCP(k) problem as a Weak-PCP(k, k) with morphisms h, g defined above.

It is known, [19] that $PCP(7)$ is unsolvable, hence we have proved that partially commutative . Weak-PCP$(7, 7)$ is unsolvable. We improve this slightly below.

We know that $PCP(2)$ is decidable (also for p.c. alphabets), this would suggest that Weak-PCP$(2, k)$ is solvable. However this suggestion is wrong.

Theorem 3.
(a) *Weak-PCP(s,r) is solvable for any s, r and noncommutative alphabets.*
(b) *Partially commutative Weak-PCP$(2, 7)$ is unsolvable.*

Proof.
(a) It easy to see that the problem Weak-PCP(s,r) for totally noncommutative alphabets is reducible to the emptiness of a language of a form:

$$(x_1 \cup \ldots \cup x_s)^+ \ \cap \ (y_1 \cup \ldots \cup y_r)^+.$$

This is a simple instance of emptiness problem for a classical regular language, hence it is obviously solvable.

(b)

Let us consider an instance of PCP(7) for lists $(u_1, u_2, \ldots u_7)$ and $(v_1, v_2, \ldots v_7)$. Assume the alphabet of words u_i, v_i is $\Sigma = \{a, b\}$. Let $\overline{\Sigma}$ be the disjoint copy of Σ, by $\overline{v}$ we mean the word v with each letter $v[i]$ changed to its copy $\overline{v[i]}$. The instance of PCP(7) is reduced to the problem Weak-PCP$(2, 7)$ as follows:

$$h(1) = a\ \overline{a},\ h(2) = b\ \overline{b}$$

$$g(i) = u_i\ \overline{v_i} \ \text{ for each } 1 \leq i \leq 7$$

Assume that the commutation relation is $\Sigma \times \overline{\Sigma} \cup \overline{\Sigma} \times \Sigma$. Then $PCP(k)$ has a solution iff Weak-PCP$(2, 7)$ has a solution for the morphisms h, g constructed above. Hence unsolvable problem PCP(7) is reduced to p.c Weak PCP(2,7). Consequently the partially commutative Weak PCP(2,7) is unsolvable. This completes the proof.

4 Weak-PCP$(1, k)$.

In this section we consider a solvable case of Weak PCP, the situation when one of the lists is of size 1. Especially simple is the case $k = 1$, i.e. the partially commutative Weak-PCP$(1, 1)$. The case of totally noncommutative alphabet is simple: for two words u, v we have

$$(\exists\ i, j)\ u^i = v^j \ \Leftrightarrow\ (\ uv = vu\ ).$$

Using projections $\pi_{a,b}$ we can reduce the p.c. case to the noncommutative case:

Observation 1
Partially commutative Weak-PCP$(1, 1)$ *for the words* u, v *is reducible to the test of* $uv \approx_I vu$, *in other words:*
(($\exists$ *(natural)* $i, j > 0$) $u^i \approx_I v^j$) $\Leftrightarrow$ ($uv \approx_I vu$).

Corollary 1. *Partially commutative Weak-PCP*$(1, 1)$ *is solvable in deterministic polynomial time.*

Theorem 4. *Weak-PCP*$(1, k)$ *is solvable.*

Proof.
Assume we have an instance of Weak-PCP$(1, k)$, given by the words $x_1, x_2, \ldots x_k$ and the word w.
In this problem we ask if there is a word $x \in \{1, \ldots, k\}^+$ and a natural m such that $h(x) \approx_I w^m$.

We can construct a reversal-bounded multicounter machine M which accepts all such strings x. Assume we have r pairs of the letters a, b which do not commute. The machine M has r counters, intially it is guessing the number m and is storing it in each counter.

Assume the i-th pair is (a_i, b_i), the machine M reads the input x on-line from left to right and using the i-th counter checks if $\pi_{a_i,b_i}(h(x)) = \pi_{a_i,b_i}(w^m)$
Then the problem Weak-PCP$(1, k)$ is reducible to emptiness of reversal-bounded multicounter machine, which is solvable due to [12]. This completes the proof.

Theorem 5. *Weak-PCP*$(1, k)$ *is NP-hard.*

Proof. The following problem called Exact Cover by 3-sets is NP-complete: given family of sets $X_i \subset U = \{1, 2, ..n\}$, where $1 \leq i \leq r$, each of cardinality 3, check if U is a disjoint union of a subfamily of these sets.

For a subset X_i let x_i be the string which is a list of elements of X_i. We can take the alphabet U, totally commutative, then the problem above is reduced to the problem if the string $z = 1\ 2\ 3\ ...n$ is equivalent (modulo permutation) to a concatenation of some of strings x_i.
W construct the instance of PCP(1,r+1) with lists:

$$w = z \cdot \#, \quad (x_1,\ x_2,\ \ldots x_r,\ x_{r+1} = \#),$$

where # is an additional symbol noncommuting with any other symbol.

Then Exact Cover by 3-sets is reduced to the problem if some concatenation of strings from the family $x_1, x_2, \ldots x_{r+1}$ is equivalent (modulo our partial commutation) to w^m, for some natural m. In this way we have a deterministic polynomial time reduction of Exact Cover by 3-sets to partially commutative PCP(1,r+1). Therefore the last problem is NP-hard.

We do not know if partially commutative $Weak\ PCP(1,k)$ is in NP, however we prove that it is in P for the lists of words over an alphabet of a constant size.

Define:

$$\Delta(\Sigma) \ = \ \{w \in \Sigma^+ \ : \ (\forall\ s1, s2 \in \Sigma)\ |w|_{s1} = |w|_{s2}\}.$$

In other words $\Delta(\Sigma)$ is the set of words over the alphabet Σ in which the number of occurrences of letters are the same. Let $L(M)$ be the language accepted by the automaton M.

The emptiness problem ($L(M) \cap \Delta(\Sigma) = \emptyset$) is called here the *diagonal membership* problem for M.

The following lemma can be shown using techniques from [5, 17]. One of these techniques is an interesting application of Euler theorem about Euler tours in multi-graphs. This allows to describe the membership problem as an integer linear program, where multiplicities are treated as variables, and the Euler condition related to indegree-outdegree of nodes can be expressed as a set of equations. This gives singly exponential upper bounds for the size of the solution.

Lemma 7.
The diagonal membership *problem for finite automata is in NP;*
If z is a shortest word in $L(M) \cap \Delta(\Sigma)$ then it is of singly exponential length (if there is any such z).

Another (quite new) algorithmic tool has been invented recently in [15], where it has been shown that the membership of a commutative word, given as a Parikh vector with coefficients written in binary, in a regular language is in P if the alphabet is of a constant size. Due to the fact from the previous lemma that the length of the shortest word is singly exponential the binary representation is polynomial. This, together with the result of [15] gives the following fact.

Lemma 8.
For a nondeterministic automaton M with input alphabet Σ of a constant size the diagonal membership problem is in P (solved by a deterministic polynomial time algorithm).

Remark.
The above problem is NP-complete for nonconstant alphabet, as pointed in [18]. However we do not use this result here.

We can use Lemma 8 to show the following fact.

Theorem 6.
Let A be the alphabet of words in the lists defining a partially commutative Weak PCP(1,k). If $|A| = O(1)$ then partially commutative Weak PCP(1,k) is in P.

Sketch of the proof.

Let r be the number of pairs of symbols a, b which do not commute. Let (a_j, b_j) be the j-th such pair and $w_{(j)} = \pi_{a_j,b_j}(w)$. We are to check if there is a sequence of indexes $i_1, i_2, \ldots i_m$ such that for some N:

$$(\forall\ 1 \leq j \leq r)\ \ \pi_{a_j,b_j}(x_{i_1} x_{i_2} \ldots x_{i_m}) \ = \ w_{(j)}^N.$$

We construct an automaton similar to the construction of a graph for testing unique decipherability of a set of words.
In our graph (the automaton M) each node is a tuple of r words. The j-th component is a prefix α (possibly empty) of $w_{(j)}$.

Let $A' = \{a_1, a_2, \ldots a_r\}$ be some additional symbols (acting as counters).
Then there is an edge labeled a_j^k from α to β iff for some x_i we have:

$$\alpha \cdot \pi_{a_j,b_j}(x_i) \ = \ w_{(j))}^k \cdot \beta,$$

where β is a prefix of $w_{(j)}$. For each component α in a given tuple we add such string a_j^k to the transition, and this is done for each component. Hence each edge of the graph of the automaton M is labeled by a string over the alphabet A' of counters.

In this way the automaton A is following some nondeterministically guessed x_i's and keeps on the edges the count of the number of copies of $w_{(j)}$. Hence it is enough to check additionally if for any two a_j, a_s we have the same number of occurrences of these symbols on some (the same for all components) nondeterministically guessed path from a source (empty prefix) to a sink (also empty prefix).
The path corresponds to the choice of a sequence $x_{i_1} x_{i_2} \ldots x_{i_m}$ and some natural nonzero N such that

$$x_{i_1} x_{i_2} \ldots x_{i_m} \ = \ w^N$$

Hence our problem is reduced to the diagonal membership problem for M. The machine M is of a polynomial size since we have only a constant number of noncommuting pairs (a_j, b_j). We omit details.

Remark.
The last theorem depends strongly on the result from the draft [15], without that result we should replace P by NP in the last theorem (this is also quite nontrivial and interesting, we are unable to do similar result for general alphabets).

Open problems:

1. Is partially commutative Weak-PCP$(2, 2)$ solvable ?
2. What about the complexity status of partially commutative Weak-PCP$(1, k)$, we showed it is NP-hard and is in P for constant sized alphabet. What about general alphabets, is it in NP ?
3. For which partially commutative alphabets I the problem Weak-PCP is solvable ? We suspect that it is solvable in case of transitively closed dependency relations $\mathcal{D}$ (the complement of $\mathcal{I}$).
4. What is the minimal k such that partially commutative $PCP(k)$ is unsolvable (in case of noncommutative alphabet the smallest known k is $k = 7$).

Acknowledgment

The authors thank J. Leroux, S. Lasota and Eryk Kopczynski for helpful comments related to Lemma 8.

References

1. M. Clerbout, M. Latteux: *Semi-commutations.* Information & Computation 73, pp. 59-74, 1987.
2. V. Diekert, G. Rozenberg (eds.): *The Book of Traces.* World Scientific, 1995.
3. A. Ehrenfeucht, J. Karhumäki,G. Rozenberg: *The (Generalized) Post Correspondence Problem with Lists Consisting of two Words is Decidable.* Theor. Comput. Sci. 21,pp. 119-144, 1982.
4. A. Ehrenfeucht, J. Karhumäki,G. Rozenberg: *On Binary Equality Sets and a Solution to the Set Conjecture in the Binary Case.* Journal of Algebra, 85,pp. 76-85, 19823.
5. J. Esparza: Petri Nets, Commutative Context-Free Grammars, and Basic Parallel Processes. Fundam. Inform. 31(1): 13-25 (1997)
6. A. Gibbons, W. Rytter: *On the Decidability of Some Problems About Rational Subsets of the Free Partially Commutative Monoids*, Theor. Ccomput. Sci. 48, pp. 329-337, 1986.
7. Vesa Halava, Tero Harju, Mika Hirvensalo: Binary (Generalized) Post Correspondence Problem. Theor. Comput. Sci. 276(1-2): 183-204 (2002)
8. T. Hajru, J. Karhumäki: Morphisms, in Hanbook of formal languages Vol. 1. G. Rozenberg, A. Salomaa, eds. Springer 1997
9. Tero Harju, Juhani Karhumäki, Daniel Krob: Remarks on Generalized Post Correspondence Problem. LNCS 1046 STACS 1996: 39-48
10. S. Holub, *Binary eqality sets are generated by two words.* Int. J. Algebra, 259 , pp. 1-42, 2003.
11. J.E. Hopcroft, R. Motwani, J.D. Ullman: *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 2001.
12. O.H. Ibarra: *Reversal-Bounded Multicounter Machines and Their Decision Problems.* Journal of the ACM 25 (1), pp. 116-133, 1978.

13. O.H. Ibarra, C.E. Kim: *A useful device for showing the solvability of some decision problems*. Proc. of the Eight Annual ACM Symposium on Theory of Computing (1976), pp. 135-140.
14. E.Kopczynski, personal communication (2009)
15. E. Kopczynski, Complexity of Problems of Commutative Grammars, draft (2009)
16. S. Lasota: Personal communication (2009)
17. J. Leroux: A Polynomial Time Presburger Criterion and Synthesis for Number Decision Diagrams. LICS 2005: 147-156
18. J. Leroux: Personal communication (2009)
19. Yu. Matiyasevich, G. Sénizergues: Decision Problems for Semi-Thue Systems with a Few Rules. LICS 1996: 523-531
20. A. Mazurkiewicz: *Concurrent Program Schemes and Their Interpretations*. Report DAIMI-PB-78, Aarhus University, 1977.
21. E. Ochmański: *Recognizable Trace Languages*. In [2], pp.167-204. World Scientific, 1995.