Post Correspondence Problem with Partially Commutative Alphabets

Barbara Klunder, Wojciech Rytter

Instytut Informatyki,
Uniwersytet Warszawski,
Warszawa

Post correspondence problem (PCP, in short):

given the set P of binary words:

$P = \{(l_1, r_1), (l_2, r_2), \dots, (l_k, r_k)\}$, gdzie: $l_i, r_i \in \Sigma^*$ $(1 \leq i \leq k)$

Is there a sequence of indices for which the system P has a solution

$$\boxed{(\exists\ w_1 w_2 \dots w_s \in (1 + 2 + \dots k)^+)}$$

$$\boxed{l_{w_1} l_{w_2} \dots l_{w_s} = r_{w_1} r_{w_2} \dots r_{w_s}}$$

The problem PCP has been posed by Emil L. Post in 1946.

It can be also formulated as a coding problem, for two codings (morphisms) $h, g \; : \; \Sigma \to (0+1)^+$ does it exist x such that $h(x) = g(x)$.

The constrained version (upper-bounding x) is NP-complete

If $|\Sigma| = 2$ then there is a deterministic polynomial time algorithm (without length restriction).

$|\Sigma| = 7$ - PCP unsolvable

$2 < |\Sigma| < 7$, solvability is open

For example if P is give by morphisms:

h(1)=0 h(2)=10 h(3)=10

g(1)=01 g(2)=0 g(3)=01

to $h(132) \ = \ g(132) \ = \ 01010$

```
0  10  10
01  01  0
```

Ale dla kodowania

h(1)=1101, h(2)=0110 h(3)=1

g(1)=1 g(2)=11 g(3)=110

shortest x such that $h(x) = g(x)$ has length 252. There is no sensible bound on the length of the solution

We introduce a version of PCP generalized to words over partially commutative alphabets. 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.

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.

The origins of partial commutativity is the theory of traces (i.e. monoids with partial commutations). Trace languages are regarded as a powerful means for description of behaviors of concurrent systems.

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 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 $\mathcal{I}$.

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

$$aaabbcab \quad \approx_I \ bbaaacba.$$

We interpret PCP as a problem about equality sets: 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:

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

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 $\text{EQ-SET}_I(h, g) \ = \ \emptyset$.

As an algorithmic tool (to show solvability) we use the algorithm testing emptiness of reversal bounded multicounter machines.

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 Ibarra's result.

Lemma 1 *The emptiness problem for reversal-bounded multicounter machines is solvable.*

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 $\quad u \;\approx_I\; w \;\Leftrightarrow\; (\;\forall\;(a, b) \notin \mathcal{I}\;)\;\pi_{a,b}(u) = \pi_{a,b}(w).$

Hence 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

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

Assume that $n = 2$ and $X = \{0, 1\}$. The problem PCP(2) is solvable as was proved by Ehrenfeucht, Karhumäki and Rozenberg in 1982. On the other hand Matiyasevich and Sénizergues showed that PCP(7) is unsolvable.

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, $|w|_s$ denotes the number of occurrences of s in w.

Lemma 4

(a) *If h and g are periodic then either*

EQ-SET$(h, g) = \emptyset$ *or* *EQ-SET*$(h, g) = \{w \in X^\star \ : \ \frac{|w|_0}{|w|_1} = k\}$ *for some fixed $k \geq 0$ or $k = \infty$.*

(b) *If h is periodic and g is not then EQ-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.

Lemma 5 *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.*

Lemma 6 *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 the following fact.

Theorem 1

Partially commutative PCP(2) is solvable

Aanother 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.

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. □

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

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

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).

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.

noindent 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, 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.*

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:

Partially commutative Weak-PCP$(1, 1)$ for the words u, v is reducible to the test of $uv\ \approx_I vu$, in other words:

$$(\ (\exists\ (\text{natural})\ i, j > 0)\ u^i\ \approx_I v^j\ )\ \Leftrightarrow (\ uv\ \approx_I vu\ ).$$

Corollary.
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. □

Theorem 5 *Assume k is a part of the input, then 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 a nodeterministic finite automaton M.

We consider the following problem for M.

(diagonal emptiness problem)

$L(M) \cap \Delta(\Sigma) = \emptyset$?

The following lemma can be shown using an Euler tour technique in multigraphs. 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 emptiness *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 We use the following fact, shown recently by Eryk Kopczynski.

Lemma 8

Assume the alphabet is of a constant size. Then the membership problem for a commutative word, given as a Parikh vector with coefficients written in binary, in a given regular language is in P.

Lemma 9

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

Proof We can transform M to an equivalent nondeterministic automaton of polynomial size such that its accepting states are sinks (there are no outgoing edges). Assume now that M is of this form. For an integer j let

$$\Sigma = \{a_1, a_2, \ldots a_r\},\ \Sigma^{(j)} = a_1^j a_2^j \ldots a_r^j$$

Change M to the automaton M', by adding for each accepting state q the loop (transition) from q to q labeled by $\Sigma^{(1)}$.

Due to Lemma 7 there is a constant c such that the length of the shortest word in $L(M) \cap \Delta(\Sigma)$ is upper bounded by $K = 2^{cn}$.

Now the diagonal emptiness problem is reduced to the problem whether $\Sigma^{(K)}$ is commutatively equivalent to some word accepted by M'. This problem is in P.

Remark. The above problem is NP-complete for nonconstant alphabet

Theorem 6 *If the alphabet A of words in the lists defining a partially commutative Weak PCP(1,k) is of a constant size then such an instance of the partially commutative Weak PCP(1,k) is in P.*

Proof.

Let r be the number of pairs of symbols (a, b) which do not commute. We have $r = O(1)$. Let (a_j, b_j) be the j-th such pair and denote:

$$\pi_j = \pi_{a_j, b_j}; \quad w_{(j)} \ = \ \pi_j(w)$$

We are to check if there is a sequence of indexes $i_1, i_2, \dots i_m$ such that:

$$(\exists\ N \geq 1)\ (\forall\ 1 \leq j \leq r) \ \ \pi_j(x_{i_1} x_{i_2} \dots x_{i_m}) \ = \ w_{(j)}^N.$$

We construct a nondeterministic automaton M similar to the construction of a graph for testing unique decipherability of a set of words.

The set of states of M is the set of r-tuples, the j-th component is some proper prefix α (possibly empty) of $w_{(j)}$.

For proper prefixes α, β of $w_{(j)}$ and a word x_i assume that

$$\alpha \cdot \pi_j(x_i) \;\; = \;\; w_{(j)}^c \cdot \beta$$

Then we write

$$next_j(\alpha, i) = \beta, \;\; count_j(\alpha, i) = c$$

If there is no such s, β then define $next_j(\alpha, i) = \emptyset$.

Let $A' = \{a_1, a_2, \ldots a_r\}$ be some additional symbols (acting as counters). For a state $q = (\alpha_1, \alpha_2, \ldots \alpha_r)$ and each $1 \leq i \leq k$ we create the transition

$$(\alpha_1, \alpha_2, \ldots \alpha_r) \Rightarrow (next_1(\alpha_1, i),\ next_2(\alpha_2, i), \ldots next_r(\alpha_r, i))$$

if $next_j(\alpha_j, i) \neq \emptyset$ for each j.

Such a transition is labeled by the string

$$a_1^{c_1} a_2^{c_2} a_3^{c_3} \ldots a_r^{c_r},$$

where $c_j = count_j(\alpha, i)$ for each j. 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 M 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 emptiness problem for M. The machine M is of a polynomial size since we have only a constant number of non-commuting pairs (a_j, b_j), hence the dimension of r-tuples is constant and the alphabet A' (of counters) is of a constant size. Due to Lemma 9 the diagonal emptiness problem for this automaton is in P.

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 when k is a part of the input, and is in P for constant sized alphabet when k is fixed . What about general alphabets, is it in NP ? Is it NP-hard in case of a fixed k ?

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 $k = 7$).