PERGAMON

Applied Mathematics Letters 16 (2003) 723–727

Applied Mathematics Letters

www.elsevier.com/locate/aml

The Post Correspondence Problem over a Unary Alphabet

P. RUDNICKI*
Department of Computing Science, University of Alberta
Edmonton, Alberta, Canada T6G 2E8
piotr@cs.ualberta.ca

G. J. WOEGINGER
Faculty of Mathematics, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
g.j.woeginger@math.utwente.nl

(Received and accepted June 2002)

Communicated by Dr. Masao Iri

Abstract—We consider the problem of finding a shortest solution for the Post correspondence problem over a unary alphabet. We show that the complexity of this problem heavily depends on the representation of the input: the problem is NP-complete if the input is given in compact (logarithmic) form, whereas it becomes polynomially solvable if the input is encoded in unary.

Keywords—Post correspondence problem, Computational complexity, NP-complete, Pseudo-polynomial time algorithm.

1. INTRODUCTION

The Post correspondence problem (PCP) was first introduced by Post in [1] where he showed that the problem is undecidable. It has become a milestone in the field of undecidability; it has been used to show that many problems about formal languages and grammars are undecidable. An instance of the PCP consists of an alphabet Σ, and a finite set $S = \{(v_i, w_i) : v_i, w_i \in \Sigma^*, i = 1, \ldots, n\}$ of n pairs of strings over Σ. The question is to decide whether there exists a (nonempty) sequence of integers $j_1, j_2, \ldots, j_\ell$ such that $v_{j_1} v_{j_2} \cdots v_{j_\ell} = w_{j_1} w_{j_2} \cdots w_{j_\ell}$. The number n of pairs is called the *size* of this instance. The smallest number ℓ providing a solution is called the *length* of the instance; if no solution exists, then the length is $+\infty$.

The PCP with size 2 is decidable [2], whereas the PCP with size 7 is undecidable [3]. The decidability of the PCP with size between 3 and 6 is still open. The PCP over an alphabet Σ with $|\Sigma| \geq 2$ is undecidable, whereas the PCP over an alphabet Σ with $|\Sigma| = 1$ (the so-called *unary* alphabet) is easily seen to be decidable. The bounded version of PCP, where we ask whether there is a solution no longer than the size of the problem, is NP-complete; see [4].

*Partially supported by NSERC Grant OGP9207.

0893-9659/03/$ - see front matter Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
PII: S0893-9659(03)00073-9

Recently, there have been some interest in constructing difficult PCP instances with relatively small size n, such that the length ℓ is big: the instance over $\Sigma = \{0, 1\}$ with these three pairs $\begin{pmatrix} 110 & 1 & 0 \\ 1 & 0 & 110 \end{pmatrix}$ has length 75, see [5], and admits two different solutions.[1]

When the size or the width (the length of the longest string) of an instance is increased the length can grow substantially: $\begin{pmatrix} 1101 & 0110 & 1 \\ 1 & 11 & 110 \end{pmatrix}$ has length 252; while $\begin{pmatrix} 111 & 011 & 10 & 0 \\ 110 & 1 & 100 & 11 \end{pmatrix}$ has length 302; see [6]. Even some small sized problems are not known to have finite length, for example $\begin{pmatrix} 110 & 1 & 0 \\ 1 & 01 & 110 \end{pmatrix}$; see [6].

In this technical note, we investigate the problem of computing the length of PCP instances over a unary alphabet $\Sigma = \{1\}$. The complexity of this problem heavily depends on the exact representation of the input: If we use a compact encoding that writes the string of length 12 as 1^{12}, then the problem is NP-complete; see Section 2. If we use a unary encoding that writes the string of length 12 as 111111111111, then the problem is polynomially solvable; see Section 3.

2. THE NP-HARDNESS PROOF

In this section, we consider the PCP over the alphabet $\Sigma = \{1\}$ where the word pairs (v_i, w_i) are given in compact form as $v_i = 1^{a_i}$ and $w_i = 1^{b_i}$ for $i = 1, \ldots, n$. Obviously, the exact ordering of a solution sequence $j_1, j_2, \ldots, j_\ell$ is irrelevant, and any permutation of a solution sequence will again yield a solution sequence. Let x_i count the number of times the integer j_i occurs in a solution sequence. Then our goal is to

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} x_i, \\ \text{subject to} \quad & \sum_{i=1}^{n} a_i x_i = \sum_{i=1}^{n} b_i x_i, \\ & \sum_{i=1}^{n} x_i \geq 1, \\ & x_i \geq 0, \qquad \text{for } 1 \leq i \leq n. \end{aligned}$$

By setting $c_i = a_i - b_i$ for $i = 1, \ldots, n$, we arrive at the following equivalent and extremely simple variant of integer programming:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} x_i, \\ \text{subject to} \quad & \sum_{i=1}^{n} c_i x_i = 0, \\ & \sum_{i=1}^{n} x_i \geq 1, \\ & x_i \geq 0, \qquad \text{for } 1 \leq i \leq n. \end{aligned}$$

The associated decision problem, which we further call PCP, is then as follows.

INSTANCE. Finite set of integers c_i, $1 \leq i \leq n$ and an integer bound B.

QUESTION. Is there a nonzero n-tuple of nonnegative integers x_i such that $\sum_{i=1}^{n} c_i x_i = 0$ and $\sum_{i=1}^{n} x_i \leq B$?

The following even-odd partition problem (EOP) is known to be NP-complete.

[1] See `http://www.informatik.uni-leipzig.de/~pcp`.

INSTANCE. Finite set of positive integers p_i, $1 \le i \le 2k$ that sum up to $2P$.

QUESTION. Is there an index subset I of $\{1,\dots,2k\}$ such that $\sum_{i\in I} p_i = P$ and $|I \cap \{2i-1, 2i\}| = 1$, for $i = 1,\dots,k$?

We present a polynomial time reduction of EOP to PCP.

The size of the corresponding PCP instance is $n = 2k+1$. For $i = 1,\dots,k$, we set $c_{2i-1} = n^n p_{2i-1} + n^{i-1}$ and $c_{2i} = n^n p_{2i} + n^{i-1}$. We set $c_{2k+1} = -n^n P - \sum_{i=1}^{k} n^{i-1}$. We claim that the constructed PCP instance has length of at most $k+1$ if and only if the instance of EOP has answer YES.

PROOF OF (IF). Let I be an index set that solves the EOP instance, and consider the following solution of PCP: set $x_i = 1$ if $i \in I \cup \{2k+1\}$, and otherwise set $x_i = 0$. Since $|I| = k$, this solution has the right length $\sum_{i=1}^{n} x_i = k+1$. To see that this solution indeed is feasible, we verify that

$$\begin{aligned}\sum_{i=1}^{n} c_i x_i &= \left(\sum_{i\in I} c_i\right) - n^n P - \sum_{i=1}^{k} n^{i-1} \\ &= \left(\sum_{i\in I} n^n p_i + n^{i-1}\right) - n^n P - \sum_{i=1}^{k} n^{i-1} = 0.\end{aligned}$$

PROOF OF (ONLY IF). Consider a feasible solution $x_1,\dots,x_n$ for the PCP instance with $\sum_{i=1}^{n} x_i \le k+1$. Then $x_{2k+1} \ge 1$ must hold, since c_{2k+1} is the only negative coefficient in the linear constraint. Next, we will prove by induction that for $j = 1,\dots,k$ we have $x_{2j-1} + x_{2j} = x_{2k+1}$.

For $j = 1$, consider the linear constraint $\sum_{i=1}^{n} c_i x_i = 0$ modulo n. All coefficients c_i except c_1, c_2, and c_{2k+1} are divisible by n; the exceptions c_1 and c_2 both are congruent 1 modulo n, and c_{2k+1} is congruent -1 modulo n. This yields

$$x_1 + x_2 - x_{2k+1} \equiv 0 (\mathrm{mod}\, n).$$

If $x_1 + x_2 \ne x_{2k+1}$, then $x_1 + x_2 \ge x_{2k+1} + n$ and this would contradict our assumption $\sum_{i=1}^{n} x_i \le k+1 < n$. For the inductive step, assume that we have proved the statement up to j. Consider $\sum_{i=1}^{n} c_i x_i = 0$ modulo n^{j+1}. All coefficients c_i with $i > 2j+2$ are divisible by n^{j+1}. Therefore, modulo n^{j+1} we have

$$\begin{aligned}0 \equiv \sum_{i=1}^{n} c_i x_i &\equiv \left(\sum_{i=1}^{j+1} n^{i-1}(x_{2i-1} + x_{2i})\right) - x_{2k+1} \sum_{i=1}^{j+1} n^{i-1} \\ &\equiv \left(\sum_{i=1}^{j} n^{i-1} x_{2k+1}\right) + n^j (x_{2j+1} + x_{2j+2}) - x_{2k+1} \sum_{i=1}^{j+1} n^{i-1} \\ &\equiv n^j (x_{2j+1} + x_{2j+2} - x_{2k+1}).\end{aligned}$$

This implies that $x_{2j+1} + x_{2j+2} - x_{2k+1} \equiv 0 (\mathrm{mod}\, n)$. If $x_{2j+1} + x_{2j+2} \ne x_{2k+1}$, then $x_{2j+1} + x_{2j+2} \ge x_{2k+1} + n$ and this would contradict our assumption $\sum_{i=1}^{n} x_i \le k+1 < n$. This completes the inductive argument.

We have established that $\sum_{i=1}^{n} x_i = (k+1) x_{2k+1}$ holds. Together with $x_{2k+1} \ge 1$ and with $\sum_{i=1}^{n} x_i \le k+1$, this yields $x_{2k+1} = 1$ and $x_{2j-1} + x_{2j} = 1$ for all $j = 1,\dots,k$. We define the index set $I = \{i : x_i = 1 \text{ and } 1 \le i \le 2k\}$. This set I constitutes a solution to the EOP instance.

THEOREM 1. *Computing the length of a compactly encoded instance of the Post correspondence problem over a unary alphabet is NP-complete.* ∎

A fully polynomial time approximation scheme (FPTAS) is an approximation algorithm that, for any given $\varepsilon > 0$, finds a feasible solution with objective value within a factor of $(1+\varepsilon)$ of the

optimal objective value. The running time of an FPTAS is polynomially bounded in the input size and in $1/\varepsilon$. See [4] for more information. The above NP-completeness proof also yields that the problem of computing the length of the PCP over a unary alphabet cannot have an FPTAS unless $P = NP$: Otherwise, we could choose $\varepsilon = 1/(2k)$ and decide in polynomial time whether the EOP instance has answer "yes". The existence of weaker approximation results for this PCP problem remains unclear.

3. THE POLYNOMIAL TIME RESULT

In this section, we consider the PCP over the alphabet $\Sigma = \{1\}$ where the word pairs (v_i, w_i) are encoded in unary. We assume that the word v_i is encoded as a string of a_i letters, and that the word w_i is encoded as a string of b_i letters, for $i = 1, \ldots, n$. Hence, the input size is $\Omega(\sum_i a_i + \sum_i b_i)$.

First let us dispose of some trivial cases: if $a_i = b_i$ for some i, then the instance has length 1. From now on we assume $a_i \neq b_i$ for all i. If $a_i > b_i$ for all i or if $a_i < b_i$ for all i, then the instance has length $+\infty$. From now on we assume that there exist indices s and t with $a_s < b_s$ and $a_t > b_t$. The length of the shortest solution to such an instance is at most

$$Z := \min_{0 \leq s,\, t \leq n} \left\{ \operatorname{lcm}\{b_s - a_s, a_t - b_t\} \left(\frac{1}{b_s - a_s} + \frac{1}{a_t - b_t} \right) \;\; a_s < b_s \text{ and } a_t > b_t \right\},$$

and hence is polynomially bounded in the input size. Finally, we set $a_{\max} = \max_i a_i$ and $b_{\max} = \max_i b_i$.

For an index i with $0 \leq i \leq n$, and for integers A and B with $0 \leq A \leq Z \cdot a_{\max}$ and $0 \leq B \leq Z \cdot b_{\max}$, we denote by $f[i; A, B]$ the length ℓ of the shortest sequence $\sigma = \langle j_1, \ldots, j_\ell \rangle$ of integers with the following properties:

- all elements in σ are from $1, \ldots, i$;
- the length of the word $v_{j_1} \cdots v_{j_\ell}$ equals A;
- the length of the word $w_{j_1} \cdots w_{j_\ell}$ equals B.

If there is no sequence of this form, then $f[i; A, B] = +\infty$. We compute all these values $f[i; A, B]$ by a dynamic programming approach. In the dynamic program, we always compute all the values $f[i; A, B]$ before any of the values $f[i+1; A', B']$ is computed.

For $i = 0$, we set $f[0; 0, 0] = 0$, and $f[0; A, B] = +\infty$ for all A and B with $A + B \geq 1$. For $i \geq 1$, we set

$$f[i; A, B] := \min_{0 \leq k} \{ f[i-1; A - k a_i, B - k b_i] + k : k \cdot a_i \leq A \text{ and } k \cdot b_i \leq B \}.$$

In this formula, the variable k counts the number of occurrences of the integer i in the corresponding sequence σ. The value of k is chosen such that the length of σ is minimized. In the very end, the length of the PCP instance can be computed as

$$\min \{ f[n; A, A] : 1 \leq A \leq Z \cdot \min\{a_{\max}, b_{\max}\} \}.$$

The running time of this dynamic program is easily analyzed: There are $O(n \cdot Z^2 \cdot a_{\max} \cdot b_{\max})$ values $f[i; A, B]$ to compute, and each such value is computed in $O(Z \cdot \max\{a_{\max}, b_{\max}\})$ time. Since Z, $a_{\max}$, $b_{\max}$, and n are polynomial in the input size, the overall running time is polynomial.

THEOREM 2. *Computing the length of a linearly encoded instance of the Post correspondence problem over a unary alphabet is polynomially solvable.* ∎

REFERENCES

1. E.L. Post, A variant of a recursively unsolvable problem, *Bulletin of the American Mathematical Society* **52**, 264–268, (1946).
2. A. Ehrenfeucht, J. Karhumäki and G. Rozenberg, The (generalized) Post correspondence problem with lists consisting of two words is decidable, *Theoretical Computer Science* **21**, 119–144, (1982).
3. Y. Matiyasevich and G. Sénizergues, Decision problems for semi-Thue systems with a few rules, *Proceedings of the 11^{th} Annual IEEE Symposium on Logic in Computer Science (LICS'1996)*, 523–531, (1996).
4. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, (1979).
5. R.J. Lorentz, Creating difficult instances of the Post correspondence problem, In *Proceedings of the 2^{nd} International Conference on Computers and Games (CG'2000)*, Springer LNCS 2063, pp. 214–228, (2000).
6. L. Zhao, Solving and creating difficult instances of Post's correspondence problem, M.Sc. Thesis, Dept. of Computing Science, University of Alberta, (2002).