

The Analysis and Solution of a PCP Instance

Jing Dong

School of Computer Science
Beijing Institute of Technology
Beijing, China
dongjing@bit.edu.cn

Abstract—In this paper, we present a heuristic rule for solving some specific PCP instances. We also analyze a PCP instance, and prove that this instance has no finite solution using our heuristic rule. Moreover, we find this instance has a unique infinite solution generated by a Fibonacci substitution.

Keywords—PCP; Post Correspondence Problem; Fibonacci substitution

I. INTRODUCTION

Post correspondence problem (PCP, for short) was first introduced by Post in 1946 [1], and he also proved that the PCP is an undecidable problem in its general form. An instance of

PCP consists of two morphisms $h, g: A^* \rightarrow B^*$, where A and B

are two finite alphabets. If $|A| = n$, then we say that the size of the instance (h, g) is n , and we denote it by $PCP(n)$. Given a

PCP instance (h, g) , if there exists a nonempty word $w \in A^*$ such

that $h(w) = g(w)$, then we call w a solution of (h, g) . In the PCP, it is asked whether or not an instance (h, g) has a solution.

Ehrenfeucht et al. [2] proved that the $PCP(2)$ is decidable. Halava and his cooperators [3] gave a simpler proof for the decidability of $PCP(2)$. On the other hand, Matiyasevich and Sénizergues proved in [4] that $PCP(7)$ is undecidable. By now the decidability of $PCP(3)$ to $PCP(6)$ is still unknown.

For a PCP instance (h, g) , if there is an infinite word $w = a_1 a_2 \dots$ with $a_i \in A$ for each $i = 1, 2, \dots$, such that for any finite prefix u of w , either $h(u)$ is a prefix of $g(u)$ or $g(u)$ is a prefix of $h(u)$, then we call w an infinite solution of the instance (h, g) . For a given instance of PCP, the problem to determine whether or not this instance has an infinite solution is called *infinite PCP* (ωPCP , for short).

In [5], Ruohonen proved that ωPCP is undecidable. Blondel and Canterini [6] proved that the ωPCP is undecidable for instances of size 105, using the undecidability of the halting problem of the Turing machine. Halava and Harju [7] used the undecidability of the termination problem of 3-rule semi-Thue systems and proved that the $\omega PCP(9)$ is undecidable. In an earlier paper we proved that the $\omega PCP(8)$ is undecidable [8].

As in [7], our proof also relies on the undecidability of the termination problem of 3-rule semi-Thue system. Currently the decidability of $\omega PCP(3)$ to $\omega PCP(7)$ is still unknown.

In this paper, we present a heuristic rule for solving some specific PCP instances. We can use this rule to prove some PCP instances have no solutions. Then, we consider a PCP instance, applying our rule to prove that it has no finite solution. Moreover, we prove this PCP instance has a unique infinite solution, and analyze the infinite solution.

II. PREVIOUS WORK

Many games that programmers are interesting are PSPACE hard or PSPACE complete. PCP is theoretically more difficult, so it has brought great challenges to the general artificial intelligence technology. There are some interesting PCP instances, see [9], [10]. In recent years, there have been some programs developed for solving PCP. Lorentz has developed programs that both solve and generate difficult instances of PCP [11]. Before searching the solution, Lorentz first filtered the instances and eliminated some instances that have no solution. The three filters he used are as following.

- Prefix filter (postfix filter)

If a PCP instance $(h, g): A^* \rightarrow B^*$ has a solution w , then

there must be a letter $a \in A$ such that either $h(a)$ is a prefix of $g(a)$, or $g(a)$ is a prefix of $h(a)$. Otherwise, no letter can be the beginning letter in the solution, and there is no solution of this instance.

Postfix filter is similar to prefix filter.

- Length balance filter

If a PCP instance $(h, g): A^* \rightarrow B^*$ has a solution w , then

the length of $h(w)$ is the same as $g(w)$. So, if for all $a \in A$, lengths of $h(a)$ are longer than $g(a)$, or for all $a \in A$, lengths of $g(a)$ are longer than $h(a)$, then there is no solution of this instance.

- Element balance filter

If a PCP instance $(h, g): A^* \rightarrow B^*$ has a solution w , then

for all $b \in B$, the number of b in $h(w)$ is the same as in $g(w)$. So, if there is a $b \in B$, for all $a \in A$, the number of b in $h(a)$ are more than in $g(a)$, or all $a \in A$, the number of b in $g(a)$ are more than in $h(a)$, then there is no solution of this instance.

III. A NEW HEURISTIC RULE

We present a heuristic rule. It is based on the fact that if a PCP instance $(h, g): A^* \rightarrow B^*$, has a finite solution w , then for all $b \in B$, b must appear in $h(w)$ and $g(w)$ with the same count. For some PCP instances with $|A| > |B|$, we can determine the proportion of number of each letter $a \in A$ in w , according to the number of each $b \in B$ in each $h(a)$ and $g(a)$. Here we assume that all the letters in A appear in the solution.

Given an instance, if we can get the proportion of number of letters, then this information may be useful to determine a lower bound for the length of solutions (if there is a finite solution). For example

$$\begin{pmatrix} 10 & 0 & 00 \\ 0 & 000 & 01 \end{pmatrix}, \quad (1)$$

and

$$\begin{pmatrix} 11 & 0 & 00 \\ 1 & 00 & 11 \end{pmatrix}. \quad (2)$$

If instance (1) has a finite solution, then in the solution, the proportion of number of letters is 2:1:2. So, we know that, the solution length is multiple of 5. Indeed, the shortest solution of this instance is of length 10.

If instance (2) has a finite solution, then in the solution, the proportion of number of letters is 2:2:1. So, the solution length of (2) is no less than 5. Indeed, the shortest solution of this instance is of length 5.

Our method can also be used to prove some instances that have no solutions. In [11], there were some PCP instances that they suspected having no solutions, but they were not able to prove. By applying our rule, some of them can be proved have no solutions. For example

$$\begin{pmatrix} 100 & 0 & 10 \\ 0 & 100 & 100 \end{pmatrix}, \quad (3)$$

and

$$\begin{pmatrix} 101 & 1 & 01 \\ 1 & 101 & 101 \end{pmatrix}. \quad (4)$$

For instance (3), to ensure that in the solution, the number of 1's in the top string are the same as in the bottom string, the proportion of number of the first and the second letter must be 1:1. But the number of 0's in the top string will be less than in the bottom string. So, instance (3) has no solution.

For instance (4), to ensure that in the solution, the number of 0's in the top string are the same as in the bottom string, the proportion of number of the first and the second letter must be 1:1. But the number of 1's in the top string will be less than in the bottom string. So, instance (4) has no solution.

IV. ANALYZING A PCP INSTANCE

Considering the following PCP instance P :

$$\begin{pmatrix} 110 & 0 & 1 \\ 1 & 111 & 01 \end{pmatrix}, \quad (5)$$

this instance can write as $P=(h_1, g_1)$, where the morphisms $h_1, g_1: \{a, b, c\}^* \rightarrow \{0, 1\}^*$ are defined by

$$\begin{aligned} h_1(a) &= 110, & g_1(a) &= 1, \\ h_1(b) &= 0, & g_1(b) &= 111, \\ h_1(c) &= 1, & g_1(c) &= 01. \end{aligned}$$

We will prove that P has no finite solution, but P has a unique infinite solution.

A. P has no finite solution

Now we use the heuristic rule in section 3 to prove the instance P has no finite solution. We can get that

Lemma 1: If P has a finite solution, then in the solution, the proportion of number of letters a, b and c must be 3:1:4.

Indeed, we can show that

Lemma 2: The letter b can't occur in solutions of P .

Proof. Suppose to the contrary that the letter b occurs in a solution w_1 of P , then w_1 can write as

$$w_1 = xby,$$

where $x \in \{a, c\}^+$ and $y \in \{a, b, c\}^*$. Comparing the lengths of $h_1(x)$ and $g_1(x)$, there are two cases, $h_1(x)$ is longer, or $g_1(x)$ is longer. Note that if the length of $h_1(x)$ is the same as $g_1(x)$, then x is a solution of P .

1) Assume that the length of $h_1(x)$ is longer.

We can write $h_1(x) = g_1(x)s$, where $s \in \{0, 1\}^+$. Note that $x \in \{a, c\}^+$, so 1 is a suffix of $g_1(x)$, and 1111 is a suffix of $g_1(xb)$. So there must be 1111 in $h_1(x)$. Because $x \in \{a, c\}^+$, $h_1(a)=110$ and $h_1(c)=1$, there must be a substring cc in x . By $g_1(c)=01$, there must be a substring 0101 in $g_1(x)$. Because $g_1(x)$ is a prefix of $h_1(x)$, there must be a substring 0101 in $h_1(x)$. Note that $h_1(a)=110$ and $h_1(c)=1$, so there can't be substring 0101 in $h_1(x)$. This is a contradiction, so the length of $h_1(x)$ can't be longer than $g_1(x)$.

2) Assume that the length of $g_1(x)$ is longer.

Since $x \in \{a, c\}^+$, we know that if $g_1(x)$ is longer than $h_1(x)$, then the number of letter c in x must be twice more than letter a . So there must be a substring cc in x , then x can write as

$$x = ucv,$$

where $u = (a^+c)^+$, $v \in \{a, c\}^*$. Note that cc is a suffix of uc , then 0101 is a suffix of $g_1(uc)$. By the definition of h_1 and g_1 , we can get $|h_1(x)| > |h_1(u)| > |g_1(u)|$, then there must be 0101 in $h_1(x)$. We have analyzed that there can't be 0101 in $h_1(x)$. We get a

contradiction which is the same as above case, so the length of $g_1(x)$ can't be longer than $h_1(x)$.

According to the analysis above, the letter b can't occur in any solution of P . \square

Lemma 1 and Lemma 2 provide that

Theorem 1: The instance P has no finite solution.

B. P has a unique infinite solution

We will prove that

Theorem 2: P has a unique infinite solution, and only letters a and c occur in this infinite solution.

Proof. We have proved that a solution of P can only contain letters a and c , and no two consecutive c . By lengths difference between $h_1(a)$ and $g_1(a)$, $h_1(c)$ and $g_1(c)$, we can get that, for any finite prefix r of a solution, the length of $h_1(r)$ must be longer than $g_1(r)$. Since there are no substring 00 in $h_1(r)$, so there always exists a letter can match continually. So P has a infinite solution.

Note that 1 is a prefix of $g_1(a)$ and 0 is a prefix of $g_1(c)$, so there is only one letter can follow r . Thus the infinite solution of P is unique. \square

Let us solve this instance. The solution should begin with the letter a , and then letters are selected one by one. The beginning of the solution is as follows:

- 110 ①
- 1
- a
- 110110 ②
- 11
- $a a$
- 1101101 ③
- 1101
- $a a c$
- 1101101110 ④
- 11011
- $a a c a$
- 11011011101 ⑤
- 1101101
- $a a c a c$
- 11011011101110 ⑥
- 11011011
- $a a c a c a$
- 110110111011101110 ⑦
- 110110111
- $a a c a c a a$
- 1101101110111011101 ⑧
- 11011011101
- $a a c a c a a c$

$$\begin{array}{l} 110110111011101101110 \\ 110110111011 \\ a a c a c a a c a \end{array} \quad \textcircled{9}$$

$$\begin{array}{l} 110110111011101101110110 \\ 1101101110111 \\ a a c a c a a c a a \end{array} \quad \textcircled{10}$$

Denote the infinite solution of P by $\hat{w}=s_1s_2s_3\dots$, where for all $i, s_i \in \{a, c\}$. Define t_i , for all $i, h_1(s_i)=g_1(s_i)t_i$. Several t_i are enumerated, see Fig. 1.

It is interesting that t_3, t_5, t_8, t_{13} are generated by a Fibonacci substitution. The substitution rule is

$$\rho: \begin{cases} 1101 \rightarrow 1101101 \\ 101 \rightarrow 1101 \end{cases}$$

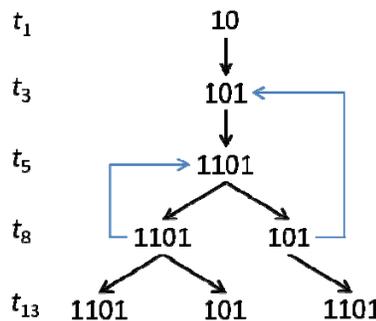


Figure 1. Fibonacci substitution sequences of t_i .

The process for solving the instance P can be viewed as continuously generating Fibonacci substitution sequences. The substitution rule is

$$\rho': \begin{cases} a \rightarrow ac \\ c \rightarrow a \end{cases}$$

Then we get the following result

Theorem 3: The infinite solution of P is $\hat{w}=\alpha_1\alpha_2\alpha_3\dots$, where $\alpha_1=a, \alpha_2=ac$, and for all $i \geq 3, \alpha_i=\alpha_{i-1}\alpha_{i-2}$.

ACKNOWLEDGMENT

We thank Prof. Liu Qinghui for his useful guidance, help and discussions. We thank the referees for helpful comments and suggestions.

REFERENCES

- [1] E. Post, A variant of a recursively unsolvable problem. Bull. Amer. Math. Soc. 52 (1946), 264–268.
- [2] A. Ehrenfeucht, J. Karhumäki and G. Rozenberg, The (generalized) Post Correspondence Problem with lists consisting of two words is decidable. Theoret. Comput. Sci. 21 (1982), 119–144.
- [3] V. Halava, T. Harju and M. Hirvensalo, Binary (generalized) Post Correspondence Problem. Theoret. Comput. Sci. 276 (2002), 183–204.
- [4] Y. Matiyasevich and G. Sénizergues, Decision problems for semi-Thue systems with a few rules. Theoret. Comput. Sci. 330 (2005), 145–169.

- [5] K. Ruohonen, Reversible machines and Posts Correspondence Problem for biprefix morphisms. *J. Inform. Process. Cybernet. EIK* 21 (1985), 579–595.
- [6] V.D. Blondel and V. Canterini, Undecidable problems for probabilistic automata of fixed dimension. *Theor. Comput. Syst.* 36 (2003), 231–245.
- [7] V. Halava and T. Harju, Undecidability of infinite Post Correspondence Problem for instances of size 9. *RAIRO–Theor. Inf. Appl.* 40 (2006), 551–557.
- [8] J. Dong and Q. Liu, Undecidability of infinite post correspondence problem for instances of size 8. *RAIRO–Theor. Inf. Appl.* (2012), Online publication.
- [9] Gurari, E.: *An Introduction to the Theory of Computation*. Computer Science Press, 1989.
- [10] Manna, Zohar: *Mathematical Theory of Computation*. McGraw Hill Inc, 1974.
- [11] R. J. Lorentz, Creating difficult instances of the post correspondence problem. *The Second International Conference on Computers and Games (CG'2000)*, Hamamatsu, Japan, 2000, 145-159.